В. В. Липаев

Очерки истории отечественной программной инженерии 1940-е – 80-е годы

DirectMEDIA

SHOUT
SHARE
IT

Annotation

Монография начинается с истории появления в нашей стране электронных вычислительных машин (ЭВМ) и программирования в 1940-е – 60-е годы. Далее изложена история проектирования и производства отечественных ЭВМ, а также средств и систем автоматизации технологических процессов производства программных продуктов в 1960-е – 80-е годы. Подробно представлена история формирования основных компонентов программной инженерии в 1960-е – 70-е годы. Внимание акцентируется на особенностях решения сложных задач по государственным заказам и на создании программных продуктов для мобильных и бортовых ЭВМ реального времени. Особое внимание уделяется истории разработки методов моделирования динамических объектов и стендов для тестирования и испытаний комплексов программ в реальном времени. Изложены методы оценивания качества программных продуктов, рисков, дефектов и ошибок при их разработке, а также история формирования требований к профессиям и квалификации специалистов программной инженерии в 1970-е – 80-е годы. Рассмотрен анализ сложности программных комплексов реального времени и распределение ресурсов ЭВМ для таких комплексов, характеристики и методы оценивания качества их компонентов. Один из разделов посвящен истории формирования в 1980-годы экономики программной инженерии, созданию средств технико-экономического анализа и экономическому обоснованию планов разработки крупных программных продуктов. Представлены реальные примеры их создания в 1960-е – 80-е годы для оборонных систем на основе методов программной инженерии. Книга предназначена для специалистов по вычислительной технике и программной инженерии, программистов, студентов и аспирантов, интересующихся историей развития, успехами и проблемами отечественной науки и техники в этой области.

- Владимир Липаев
  - Введение
  - Глава 1. История появления в стране вычислительной техники и программирования в 1940-е – 60-е годы
    - 1.1. Начало истории отечественной вычислительной техники в 1940-е – 60-е годы
    - В середине 50-х годов в оборонных отраслях
    - 1.2. Начало истории отечественного программирования в 1950-е – 60-е годы

- 1.3. Первые комплексы программ для оборонных систем в 1950-е – 60-е годы
- 1.4. Организация подготовки первых программистов в 1950-е – 60-е годы

○ Глава 2. История отечественной вычислительной техники в 1950-е – 70-е годы
- 2.1. История семейства стационарных универсальных вычислительных машин «Урал» в 1960-е – 70-е годы
- 2.2. История стационарных универсальных, высокопроизводительных ЭВМ в 1960-е – 70-е годы
- 2.3. История стационарных, специализированных ЭВМ реального времени в 1970-е – 80-е годы
- 2.4. История единого семейства (ЕС) ЭВМ в 1970-е – 80-е годы
- 2.5. История семейства малых универсальных ЭВМ в 1970-е – 80-е годы
- 2.6. История мобильных и бортовых специализированных ЭВМ в 1960-е – 80-е годы
- 2.7. История семейства «Аргон» с архитектурой и системами команд ЕС ЭВМ в 1970-е – 80-е годы

○ Глава 3. История технологических систем для производства программных продуктов в 1960-е – 80-е годы
- 3.1. История технологической программной системы ЭВМ «Урал»
- 3.2. История операционной программной системы
- 3.3. История разработки основной операционной системы ОС ЕС
- 3.4. История технологической программной системы
- 3.5. История технологии программирования для специализированных ЭВМ реального времени на БЭСМ-6 в 1970-е годы
- 3.6. История технологии программирования для бортовых ЭВМ реального времени на ЕС ЭВМ в 1980-е годы
- 3.7. Первая отечественная монография по проектированию крупных программных продуктов

○ Глава 4. История формирования основных

компонентов программной инженерии в 1960-е – 70-е годы
- 4.1. Особенности крупных заказных функциональных задач для ЭВМ в 1950-е – 60-е годы
- 4.2. Особенности ЭВМ, ориентированных на задачи в реальном времени
- 4.3. Создание операционных систем и глобальной телекоммуникационной сети ЭВМ реального времени в 1965-е годы
- 4.4. Моделирование динамических объектов для тестирования и испытаний комплексов программ в реальном времени в 1960-е – 80-е годы
- 4.5. Методы оценивания качества, рисков, дефектов и ошибок в программных продуктах в 1960-е – 80-е годы
- 4.6. Формирование профессий и квалификаций специалистов программной инженерии в 1970-е – 80-е годы

○ Глава 5. История развития методов программной инженерии для систем реального времени в 1970-е – 80-е годы
- 5.1. Анализ сложности и распределения ресурсов ЭВМ для программных комплексов реального времени в 1970-е годы
- 5.2. Характеристики и методы оценки качества компонентов и комплексов программ реального времени – 1975-е годы
- 5.3. Методы оценивания и обеспечения корректности и надежности программных продуктов – 1975-е годы
- 5.4. История формирование экономики программной инженерии в 1980-е годы
- 5.5. Сбор и обобщение экономических характеристик о жизненном цикле программных комплексов реального времени в 1980-е годы
- 5.6. Технико-экономический анализ и обоснование разработки комплексов программ – 1985-е годы
- 5.7. Методики технико-экономического обоснования производства комплексов программ – 1985-е годы

○ Глава 6. Примеры создания заказных систем на основе

программной инженерии в 1960-е – 80-е годы
- 6.1. Технологии программной инженерии для крупных комплексов программ системы ПВО в 1960-е – 80-е годы
- 6.2. Создание и внедрение программных продуктов систем противокосмической обороны и морской космической разведки
- 6.3. Крупные комплексы программ реального времени в системах предупреждения о ракетном нападении
- 6.4. Создание бортовых программных комплексов стратегических ракет и космических аппаратов
- 6.5. Космические аппараты информационных систем
- 6.6. Бортовые программные комплексы авиационных систем в 80-е годы

○ Заключение
○ Приложение
○ Литература

Владимир Липаев
Очерки истории отечественной программной инженерии в 1940-е -80-е годы

Введение

Истории развития отечественной вычислительной техники в Советском Союзе в начале 1950-х годов сопутствовали две противоречивые тенденции, существенно повлиявшие на прогресс науки и техники в стране. С одной стороны, началась идеологическая атака в средствах массовой информации на «буржуазную лженауку – кибернетику». С другой стороны, постановления Правительства активно стимулировали разработки вычислительных машин в нескольких организациях [3, 4] страны.

В 1948 году была издана книга американского математика Норберта Винера «Кибернетика или управление и связь в животном и машине», которая в СССР попала на полки с ***секретными изданиями.*** Ее автор высказал идеи, не согласующиеся с официальными философскими доктринами, пропагандировавшимися в советском обществе. Для Винера было ясно, что многие концептуальные схемы, определяющие поведение живых организмов при решении конкретных задач, идентичны схемам, характеризующим процессы управления в сложных технических системах. Он был убежден, что социальные модели управления в человеческом обществе и модели управления в экономике могут быть проанализированы на основе тех же общих методов и положений, которые разработаны в области управления техническими системами, созданными людьми. Эти ***крамольные идеи*** не могли стать достоянием советских граждан, которым внушался тезис марксистской философии о несводимости «высших форм» существования материи к «низшим формам», используемым в технике.

В журнале «Вопросы философии» в марте 1950 года критике были подвергнуты некоторые теоретические положения математической логики, противоречившие, по мнению авторов статьи, догмам материализма [4]. Они не скупились на резкие высказывания: «Речь идет не о том, чтобы ликвидировать математическую логику, а о том, чтобы отсечь реакционную тенденцию извращения ее, отражающую идеологию враждебных нам классов». После математической логики настала очередь массированной атаки на те направления в физиологии, которые нарушали «чистоту учения И.П. Павлова», объявленного марксистскими философами венцом учений о поведении животных, и той части поведения человека, которая регулировалась его центральной нервной системой.

В 1953 году наступила очередь агрессивных статей

философов о кибернетике. Вершиной наступления на кибернетику стала статья, напечатанная в журнале «Вопросы философии» в 1953 году. Она была помещена в разделе «Критика буржуазной идеологии» и называлась «Кому служит кибернетика» (автор скрылся под псевдонимом). Все, что касалось развития вычислительной техники как таковой, когда вычислительные машины уподоблялись очень быстро работающим арифмометрам, ***объявлялось полезным и нужным для социалистического отечества.*** В подобном качестве вычислительные машины ничем не отличались от устройств, создаваемых человеком для облегчения своего труда. Однако, когда речь заходила об использовании этих машин для моделирования различных процессов или для символьных преобразований, то натренированный на поиске идеологического криминала ум борца за чистоту марксистско-ленинского учения немедленно ***подавал сигнал опасности***: «По мнению Винера, деятельность вычислительных машин дает ключ к познанию самых разнообразных природных и общественных явлений. ***Эта в корне порочная идея послужила Винеру основанием для создания новой «науки» – кибернетики».***

Вычислительные машины не могут внести качественно новую струю в процесс познания окружающего мира. Чтобы эта мысль дошла до всех читателей статьи, автор формулирует ее еще раз: «Теория кибернетики, пытающаяся распространить принципы действия вычислительных машин новейшей конструкции на самые различные природные и общественные явления без учета их качественного своеобразия, является механицизмом, превращающимся в идеализм. Это пустоцвет на древе познания, возникший в результате одностороннего и чрезмерного раздувания одной из черт познания». Набор ярлыков для кибернетики (пустоцвет, лженаука, идеологическое оружие империалистической реакции, ***порождение лакеев империализма)*** свидетельствовал, что никакой патриотически настроенный ученый в СССР не может заниматься столь одиозной наукой. В «Кратком философском словаре» (1954 год) в статье «Кибернетика» эта наука была определена как ***«реакционная лженаука,*** возникшая в США после второй мировой войны, получившая широкое распространение и в других капиталистических странах; форма современного механицизма».

Практические задачи (и прежде всего, задачи укрепления обороноспособности страны) требовали не прекращения работ в этой области, а наоборот, расширения и активизации этих исследований. Это понимали даже партийные чиновники из оборонного отдела и отдела науки ЦК КПСС. Когда один из

первых отечественных специалистов по применению вычислительных машин в военной области А.И. Китов, математик А.А. Ляпунов и известный своими теоретическими работами, связанными с созданием атомной бомбы, математик С.Л. Соболев объединились как авторы статьи «Основные черты кибернетики» и принесли ее в журнал «Вопросы философии», то: «Как ни странно, редколлегия спорить не стала» и в начале 1955 года статью опубликовали [4]. В это время А.И. Китов и А.А. Ляпунов организовали серию выступлений на научных семинарах в академических институтах, высших учебных заведениях и в других организациях, в которых методы кибернетики могли бы принести практическую пользу и ***наступление на кибернетику оголтелых философов к концу 50-х годов постепенно увяло.***

У истоков развития кибернетики (информатики) в СССР стояли сотрудники Академии наук и различных, именуемых «закрытыми» ведомств и предприятий, в большинстве своем связанные с оборонной техникой. Первые книги в области кибернетики, вычислительных машин и программирования, выпущенные уже во второй половине ***50-х годов*** без грифа секретности, были написаны военными. Если бы не ***активная наступательная позиция военных,*** поддержанная членами Академии наук СССР, то идеологические концепции, охраняемые представителями ***консервативной философской и партийной элиты,*** возможно, задержали бы развитие информатики на несколько лет, как это случилось с генетикой и другими науками [3, 4].

С середины 40-х годов началась и на протяжении нескольких десятилетий двадцатого века продолжалась ***история практически независимого от Запада, оригинального развития отечественной вычислительной техники и программирования, науки и техники в этой области.*** Созданные нашими учеными, инженерами и рабочими сложные вычислительные системы для различных сфер применения многие годы находились ***на уровне мировых достижений и не содержали зарубежных компонентов.*** Этот ***путь самостоятельного развития отечественных ЭВМ и программирования,*** в том числе, в секретных (закрытых) областях применения, практически неизвестен современному поколению специалистов в области информатики. Он почти не отражен в научно-технической, исторической литературе, а также в учебных курсах вузов. История развития вычислительной техники в нашей стране имеет ***много достойных результатов*** и их не следует прятать в архивах. Целесообразно сделать их доступными для установления справедливых оценок достижений и просчетов, для извлечения

современными специалистами уроков из накопленного опыта и дискуссий предшествующих десятилетий, с учетом бурного развития этой области. Автор книги полагает, что это может быть интересно и полезно для широкого круга специалистов, поэтому эту историю целесообразно обобщить и ***объективно отразить в средствах массовой информации.***

Уже на начальных этапах развития вычислительной техники и программирования в конце 1950-х годов произошло разделение в этой сфере на ***два почти независимых направления:*** для применения в науке и народном хозяйстве и для оборонных систем. ***Первое направление*** вначале составляли относительно небольшие программы, создаваемые одиночками или небольшими коллективами (3–5) специалистов, преимущественно для получения конкретных результатов при автоматизации научных исследований, простых технологических процессов и/или для анализа результатов вычислений самими разработчиками программ. В 60-е годы десятки и даже сотни ЭВМ с оригинальной архитектурой и системами команд, почти кустарно изготавливались в единичных экземплярах в ***вузах, научных и исследовательских организациях*** и не имели перспективы их массового производства. Для многих видов относительно несложных программ не было необходимости в регламентировании и автоматизации жизненного цикла для длительного применения и сопровождения множества версий таких программ.

Это приводило к созданию для них простейших операционных систем и технологических средств для пользователей с очень ограниченными возможностями. Их разработчики не применяли регламентирующих, нормативных документов, вследствие чего жизненный цикл таких продуктов по структуре, содержанию, качеству и стоимости основных процессов «творчества» имел ***непредсказуемый характер.*** Многие авторы таких систем не стремились создать упорядоченные методы и средства массового проектирования и производства программных продуктов гарантированного качества.

Пеструю историю их создания и развития в 60-е годы трудно отразить системой методов и средств. Только в 70-е годы сформировались несколько базовых семейств ЭВМ, для которых были созданы операционные системы и технологические средства, обеспечивающие проектирование и производство больших прикладных комплексов программ. В 70-е годы начало меняться направление основных усилий разработчиков программ для ЭВМ, происходит переход от методов и процессов процедурного программирования небольших компонентов для

решения частных задач к ***проектированию крупных комплексов программ для промышленных и административных систем реального времени.*** Увеличение ресурсов ЭВМ и их доступности стимулировало интенсивное расширение сфер применения и возрастание размеров создаваемых комплексов программ. Быстро увеличивались производительность, объем памяти и надежность ЭВМ, что позволяло повышать сложность выполняемых ими функций и решаемых с их помощью задач, расширять сферы их использования в науке, системах управления и в промышленности. Однако отсутствовали стимулы для объединения методов и средств создания и сопровождения программ и, как следствие, для формирования систематизированного набора положений ***«программной инженерии».***

Второе направление с середины 60-х годов составляли крупные ***заказные (создаваемые по заказу государства) комплексы программ реального времени для сложных оборонных систем*** управления и обработки информации. Такие комплексы создавались большими коллективами специалистов, преимущественно в оборонной промышленности, оформлялись в виде ***программных продуктов*** с гарантированным качеством. Эти комплексы программ являлись компонентами систем, реализующими их основные функции и содержащими предпосылки для последующего развития и изменений. Методология управления проектами программных продуктов зависела от многих факторов: от персонала, технических, организационных, договорных требований и сложности функций. Организованная и контролируемая коллективная разработка при строгом учете и контроле каждого изменения являлась основой эффективного, поступательного развития каждой крупной вычислительной системы ***методами программной инженерии.*** Руководством страны особенно активно стимулировалось развитие таких комплексов программ для оборонных систем. В предлагаемой монографии внимание ***акцентируется на истории и крупных достижениях технологии программирования в оборонной промышленности страны, наименее известных современным специалистам.***

В ***1960-е – 80-е годы в оборонной сфере*** были сосредоточены огромные ресурсы науки и промышленности, работали сотни тысяч разработчиков сложных комплексов программ, в несколько раз больше, чем в гражданских отраслях. Концентрация специалистов, стимулирование их труда, естественно, давали результаты. Это, в частности, отражалось на активном ходе работ в области создания ***крупных интеллектуальных программных***

продуктов и технологических систем программной инженерии для повышения эффективности процессов разработки и оценки качества комплексов программ оборонных систем. Однако методологические и технологические достижения в этой области передовых предприятий оборонной промышленности ***оставались секретными,*** не отражались в открытых публикациях и зачастую были не известны специалистам даже близких по функциям и задачам предприятий.

Разработки комплексов программ для оборонных систем с самого начала ***отличались организованностью и тесным взаимодействием с заказчиками*** таких систем. При этом требования заказчиков к функциям и качеству программных продуктов ***хронически превышали возможности разработчиков*** и ресурсы доступных вычислительных машин. Очень быстро расширялись функции комплексов программ и, соответственно, потребности в размерах памяти и производительности ЭВМ, на которых они использовались. Это заставляло разработчиков программ создавать и использовать ***эффективные алгоритмы решения поставленных задач и методы программирования.*** Это стимулировало совершенствование тех и других, и необходимость формализации технологий применявшейся тогда программной инженерии. Расширение размеров и функций, создаваемых программных продуктов, а также уровень автоматизации их проектирования и программирования следовали непосредственно за увеличением доступных ресурсов вычислительных машин. Одновременно повышалась производительность труда специалистов и качество программного продукта.

Развитие вычислительной техники в 1960-ые годы происходило в Советском Союзе очень высокими темпами. Предприятия активно оснащались различными ЭВМ. Кульминационной точкой в истории отечественной вычислительной техники стало создание С.А. Лебедевым в 1967-ом году ЭВМ БЭСМ-6. Именно эта машина впервые поразила весь мир невероятной для того времени производительностью – один миллион операций в секунду. Машина БЭСМ-6 сильно опередила свое время, начав развитие второго поколения ЭВМ. Она вобрала в себя много оригинальных идей, подобного класса в мире тогда не было. Эта машина широко использовалась в системах автоматизации проектирования для моделирования сложнейших физических процессов и процессов управления как инструментальная машина для разработки ***крупных программных продуктов оборонных систем*** на базе различных мобильных и бортовых ЭВМ. Она оставалась востребованной рекордно долгое время, более тридцати лет – последний

экземпляр БЭСМ-6 прекратили использовать только на рубеже 21-го века.

К середине 80-х годов в стране было создано ***около 300 типов и более десяти семейств оригинальных ЭВМ,*** в основном, для оборонной техники [10, 11]. Однако большое число проектов оставалось на уровне экспериментальных образцов. Они не определяли вычислительный потенциал страны и не отражены в данной книге. Последующее изложение ориентировано на ограниченное число типов ЭВМ, сыгравших наиболее важную роль в отечественной истории развития вычислительной техники и программной инженерии. Большинство из них были оснащены отечественными операционными системами, трансляторами и отладчиками. Инструментальные средства проектирования и производства программных продуктов, естественно, были ориентированы на определенные типы аппаратуры ЭВМ и в большинстве случаев определялись используемыми вычислительными ресурсами, функциями и областями их применения. ***Средства программной инженерии*** могли реализоваться только при достаточно больших ресурсах ЭВМ. Это определило их появление и активное применение, начиная с середины 60-х годов для оборонных систем.

В 1980-е годы начинает формироваться и систематизироваться программная инженерия для проектирования крупных комплексов программ административных, гражданских сфер народного хозяйства. В эти годы происходит переход к массовому производству сложных комплексов программ высокого качества и к подготовке специалистов для поддержки жизненного цикла таких программных продуктов. На многих предприятиях началась осваиваться методология программной инженерии. ***Завершалась эпоха самостоятельного развития ряда поколений отечественной вычислительной техники и операционных систем*** для широкого применения в народном хозяйстве. Оригинальные отечественные разработки в этой области сохранялись, в основном, в оборонных отраслях промышленности. В то же время проявилась тенденция к сокращению разнообразия архитектур, к унификации мобильных, бортовых и наземных ЭВМ оборонного назначения, к их сближению с архитектурами универсальных вычислительных машин.

Этап оригинального развития вычислительной техники в СССР пошел на спад в конце 1970-х годов, когда было принято решение о переходе к производству и использованию ЭВМ, которые являлись прототипами моделей западных образцов – IBM

360 и PDP. Руководители высшего управленческого уровня ***не понимали*** уже освоенных в стране ***методов и технологий программной инженерии, направленных на создание сложных комплексов программ***. Они видели, что на Западе имеется программный продукт, который ***«.без особого труда»*** можно ***нелегально*** копировать и использовать, если наладить производство аппаратуры ЭВМ с соответствующей архитектурой. В угоду ***приоритету ЕС ЭВМ*** были ***оборваны и прекратили существование*** отечественные линии проектирования и производства семейств универсальных вычислительных машин, в частности, БЭСМ-6 и «Урал». Освоение зарубежных операционных систем, СУБД, прикладных и технологических программ для этих типов машин ***подорвало оригинальную, отечественную школу программирования*** и сориентировало ее на заимствование и адаптацию готовых, как правило, неизвестного качества, зарубежных программ. Такая тенденция стала в то время доминирующей. Проектирование и производство оригинальных советских ЭВМ – это ***успехи прошлого нашей страны.*** Вместе с тем, это свидетельство широких возможностей наших ученых и специалистов, которые, к сожалению, почти утрачены в настоящее время.

Однако для ряда специальных, критических сфер применения оборонных систем (например, мобильных, бортовых в авиационной, ракетной и космической технике), ***сохранили актуальность разработка и использование унаследованных архитектур*** специализированных ЭВМ реального времени на новой элементной базе. К ним предъявлялись особенно высокие требования к качеству, надежности, габаритам, климатическим характеристикам. Это обусловливало значительные особенности их архитектур и систем команд, в том числе для сохранения и модернизации ранее разработанных и эксплуатируемых крупных функциональных программных продуктов оборонных систем.

В конце 80-х годов началась ***очередная смена поколения*** вычислительной техники и активное освоение различных зарубежных персональных ЭВМ и серверов с резким увеличением доступных пользователям ресурсов. Эти годы стали ***переломными для истории оригинального развития*** отечественной вычислительной техники и программирования. Широкий поток в страну зарубежных персональных ЭВМ драматически отразился на создании отечественных средств автоматизации программирования и программной инженерии. Изобилие разнообразных программных продуктов для персональных ЭВМ переориентировало отечественных специалистов на их освоение и применение. Только в некоторых направлениях

инструментальных средств программной инженерии (например, тестирование, компиляторы) продолжались исследования и создавались отдельные, принципиально новые технологические средства высокого качества.

Изменение акцентов в деятельности многих отечественных специалистов, связанных с переходом от индивидуального программирования небольших компонентов ***к коллективному созданию и применению методов и технологий программной инженерии,*** происходило в 1960-е – 80-е годы. В различных, (в основном, оборонных) областях применения программных продуктов появилась потребность в обеспечении экономической эффективности процессов проектирования и производства крупных комплексов программ. Для этого начали использоваться наиболее совершенные методы управления проектами, системная автоматизация процессов на всех этапах жизненного цикла комплексов программ для повышения производительности и качества результатов труда участвующих в этих процессах специалистов. Управленческие и технические проверки, анализ качества результатов выполнения промежуточных работ и созданных компонентов, проверки корректности их взаимодействия должны были обеспечивать заказчикам, руководителям и всем разработчикам более высокую степень уверенности в ***достижении требуемого конечного результата, и гарантии качества программного продукта.***

Программные продукты, созданные для оборонных систем, стали ***интеллектуальной их основой,*** от качества которой зависят выполняемые ими функции и последствия их применении. К сожалению, руководители предприятий и исполнители проектов до сих пор должным образом не осознают того факта, что многие дефекты, нештатные ситуации и ***катастрофы сложных критических систем*** являются следствием низкого качества поддерживающих их функционирование программных продуктов, напрямую зависят от недостатков при применении методов и средств программной инженерии.

Публикации по истории отечественной программной инженерии в последнее время рассекречены, многие из них представлены в Интернете. Однако они не отличаются систематичностью изложения и разнообразием. Для устранения этих недостатков целесообразно привлекать, прежде всего, тех специалистов, которые в те далекие годы, непосредственно участвовали в проектах и исследованиях, в том числе работая в многочисленных закрытых учреждениях. Большинство таких специалистов предпочитали заниматься

основной, профессиональной деятельностью в ущерб анализу истории и особенностей, протекающих в те или иные годы процессов. Основные публикации по истории отечественной вычислительной техники сосредоточены на особенностях и характеристиках аппаратуры ЭВМ и только немного внимания, и очень неравномерно, уделяют созданию комплексов функциональных программ, инструментальным средствам и особенностям автоматизации программирования, в которых изредка упоминаются некоторые компоненты программной инженерии. Это определялось отсутствием необходимости создания крупных комплексов программ в различных гражданских отраслях народного хозяйства.

При подготовке монографии активно использовались доступные публикации, среди которых наибольшее влияние на ее содержание оказали, прежде всего, работы некоторых ведущих отечественных специалистов, представленные в списке литературы [1 – 8], а также материалы в Интернете [11] и Компьютерном музее [10]. Материалы по развитию программной инженерии при создании систем контроля космического пространства представили Генеральный конструктор ЦНИИ Комета, доктор технических наук, профессор Виктор Порфирьевич Мисник и доктор технических наук, профессор Владимир Федорович Гребенкин, за что автор им весьма благодарен. В предлагаемой монографии, естественно, отразились профессиональные интересы, опыт и публикации автора [16–22], более 30 лет работавшего в оборонной промышленности над крупными проектами комплексов программ.

Глава 1. История появления в стране вычислительной техники и программирования в 1940-е – 60-е годы

1.1. Начало истории отечественной вычислительной техники в 1940-е – 60-е годы

Развитием промышленности по производству средств вычислительной техники правительство и руководящие органы СССР начали серьезно заниматься практически сразу же после окончания Великой Отечественной войны, считая эту задачу одной из основных для народного хозяйства [1, 2, 3]. Поручение правительства по подготовке мероприятий, связанных с развитием вычислительной техники, было дано в период острой необходимости в капитальных вложениях для подъема, разрушенного войной народного хозяйства, одновременно с философской полемикой в печати о роли «буржуазной лженауки» кибернетики. Работы, имевшие для страны большое значение, как это было принято, поручались сразу нескольким организациям. Результатом выполнения этих поручений было ***постановление правительства*** 1948-го года, предусматривавшее создание Института точной механики и вычислительной техники (ИТМ и ВТ) АН СССР и двух отраслевых организаций: НИИсчетмаш и СКБ-245, а также расширение существующей производственной базы и выделение необходимых для этого средств. Кроме того, в ряде организаций АН СССР и различных ведомств: в лаборатории электросистем Энергетического института им. Г.М. Кржижановского в Москве; в лаборатории вычислительной техники Института математики АН УССР в Киеве (позже ВЦ АН УССР); в Ереванском институте математических машин; в

Пензенском институте управляющих вычислительных машин, активно развивалась теория и началась разработка вычислительных машин. Однако к началу 50 – х годов в стране имелись только небольшие производственные мощности по выпуску счетных и счетно-перфорационных машин, электронная вычислительная техника только зарождалась, а производственные мощности по элементной базе для нее были близки к нулю.

Послевоенные годы стали отправной точкой в истории создания первых советских ЭВМ. В 1948 – 1951-е годы в Киеве в лаборатории моделирования и вычислительной техники Института электротехники АН УССР под руководством Сергея Алексеевича Лебедева была создана первая советская малая электронная счетная машина (МЭСМ) – прототип современных ЭВМ. Созданием МЭСМ в кельях бывшего монастыря «Феофания» было положено ***начало развитию отечественной***

вычислительной техники. К концу 1950 года монтаж действующего макета первой отечественной ЭВМ был завершен, и началась его проверка путем решения тестовых и ряда простейших народно-хозяйственных задач. А уже через два года в Москве, в Институте Точной Механики и вычислительном центре АН СССР, директором которого стал С.А. Лебедев, на базе разработанной в Киеве модели создается БЭСМ – большая электронная счетная машина. Машины данной серии становятся одними из лучших в США и Европе! Сегодня в это сложно поверить, однако быстродействие БЭСМ, а также возможность выполнить сложные математические операции подтверждало высокий уровень развития науки и технологий в Советском Союзе и открывало широкие перспективы для новых открытий и достижений.

Параллельно с С.А. Лебедевым по постановлению правительства над созданием электронно-вычислительных машин проводили работы и другие ученые – с 1948-го года в конструкторском бюро № 245, возглавляемом М.А. Лесечко, шла разработка цифровой вычислительной машины, получившей название «Стрела». Скорость ее работы составила две тысячи операций в секунду, что в пять раз уступает быстродействию БЭСМ. «Стрела» впервые стала выпускаться серийно.

Важнейшим звеном в истории советской вычислительной техники стали созданные группой инженеров под руководством И.С. Брука машины «М1». Данная машина отличалась невысоким быстродействием, но ее важным преимуществом были небольшие габаритные размеры, что делало ее применение удобным в любых помещениях. Впоследствии разработки И.С. Брука были усовершенствованы, и в 1953-ем году машина «М2», скорость работы которой составляла уже 2 тысячи операций с минуту, сочетала в себе все преимущества советских ЭВМ.

Три электронно-вычислительные машины на лампах – БЭСМ, «Стрела» и «М2» относятся к советским ЭВМ ***первого поколения*** (рис. 1).

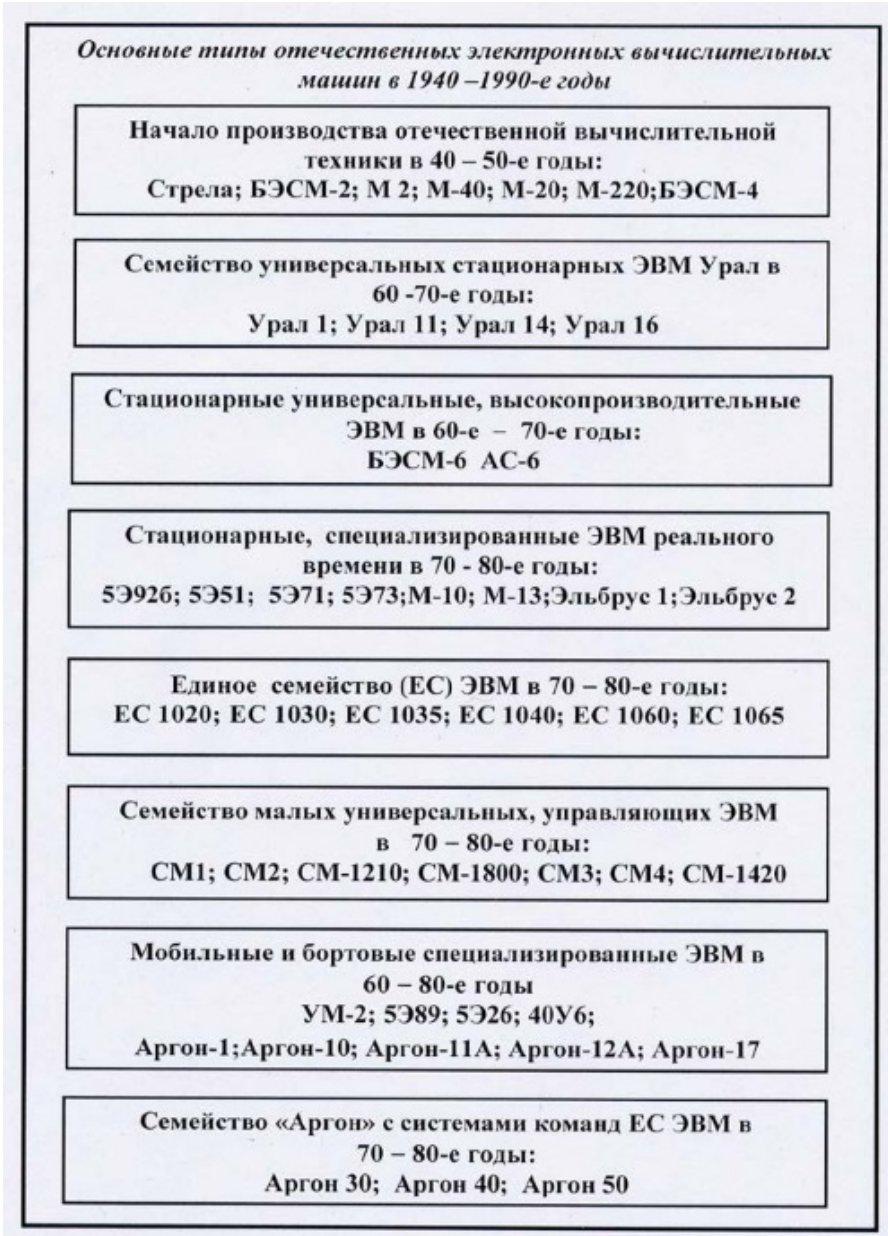
Основные типы отечественных электронных вычислительных машин в 1940 –1990-е годы
Начало производства отечественной вычислительной техники в 40 – 50-е годы:
Стрела; БЭСМ-2; М 2; М-40; М-20; М-220;БЭСМ-4
Семейство универсальных стационарных ЭВМ Урал в 60 -70-е годы:
Урал 1; Урал 11; Урал 14; Урал 16
Стационарные универсальные, высокопроизводительные ЭВМ в 60-е – 70-е годы:
БЭСМ-6 АС-6
Стационарные, специализированные ЭВМ реального времени в 70 - 80-е годы:
5Э92б; 5Э51; 5Э71; 5Э73;М-10; М-13;Эльбрус 1;Эльбрус 2
Единое семейство (ЕС) ЭВМ в 70 – 80-е годы:
ЕС 1020; ЕС 1030; ЕС 1035; ЕС 1040; ЕС 1060; ЕС 1065
Семейство малых универсальных, управляющих ЭВМ в 70 – 80-е годы:
СМ1; СМ2; СМ-1210; СМ-1800; СМ3; СМ4; СМ-1420
Мобильные и бортовые специализированные ЭВМ в 60 – 80-е годы
УМ-2; 5Э89; 5Э26; 40У6;
Аргон-1;Аргон-10; Аргон-11А; Аргон-12А; Аргон-17
Семейство «Аргон» с системами команд ЕС ЭВМ в 70 – 80-е годы:
Аргон 30; Аргон 40; Аргон 50

Рис. 1.

Все эти разработки имели существенные недостатки – высокая степень энергопотребления и небольшая оперативная память требовали совершенствования, но и западные машины того времени не превосходили советские ЭВМ по своим эксплуатационным характеристикам.

В середине 40-х годов в США был опубликован документ под названием ***«Архитектура, фон Неймана».*** В нем великий физик

и математик Джон фон Нейман (John von Neumann) описал вычислительную систему, в которой процессорный модуль отделен от устройства хранения данных. Вскоре был создан, а затем и усовершенствован первый американский компьютер ENIAC. Его установили в Университете штата Пенсильвания, США, и начали использовать для решения научных задач. Тогда же в Англии появились первые управляемые программами ЭВМ. Ученым уже было известно, что американские и британские коллеги достигли определенных успехов, но «холодная война» наложила ограничения – исследования заморских умов нашим конструкторам были недоступны.

При создании проекта МЭСМ в 1947-м году С.А. Лебедевым были независимо от работ Дж. фон Неймана сформулированы аналогичные ***основные принципы построения архитектуры электронных вы числительных машин***:

• в состав ЭВМ должны входить арифметическое устройство, память, устройство управления и устройство ввода-вывода;

• программа в машинных кодах должна храниться в той же памяти, что и числа;

• для представления чисел и команд должна применяться двоичная система счисления;

• вычисления должны выполняться автоматически в соответствии с программой, хранящейся в памяти;

• логические операции должны выполняться наряду с арифметическими операциями;

• память машины должна быть организована по иерархическому принципу.

Основой высокой эффективности деятельности С.А. Лебедева являлось понимание основополагающих принципов развития столь сложного направления человеческой деятельности, как электронная вычислительная техника, глубокий теоретический анализ выполняемых проектов [2]. Отсюда чрезвычайно высокие требования к главному конструктору и разработчикам выполняемого проекта. С.А. Лебедев тщательно обдумывал все аспекты проблемы и в результате не имел практически ни одного проекта «в корзину». Все его разработки – более полутора десятков проектов ЭВМ, были внедрены в серийное производство, из которых две трети для задач обороны страны.

С.А. Лебедев очень точно определил ***направление развития вычислительной техники.*** Ее передовым фронтом он считал высокопроизводительные вычислительные системы. Сергей Алексеевич отстоял основное направление работы ИТМ и ВТ – высокопроизводительные вычислительные системы, несмотря на то, что впоследствии Институту предлагали главную роль в

стране по разработке вычислительной техники на базе прототипов ЕС ЭВМ. Он считал, что развитие вычислительной техники определяют сверхвысокопроизводительные системы и страна должна иметь ***самостоятельное направление в этой области.***

С 1953-го года в стране был налажен ***серийный выпуск вычислительных машин.*** Первой в серию пошла ЭВМ «Стрела», созданная в СКБ-245 под руководством Ю.Я. Базилевского. Основные характеристики ЭВМ «Стрела»:

• ЭВМ была разработана на обычных для того времени радиолампах общим количеством ~ 6000 штук;

• быстродействие 2000 операций в секунду, тактовая частота 50КГц, команды трехадресные;

• оперативная память содержала 2048 ячеек, ячейка, в которой размещались трехадресная команда или число, содержала 43 двоичных разряда, оперативная память была выполнена на электронно-лучевых трубках (ЭЛТ), каждый разряд запоминался на отдельной трубке;

• общая потребляемая мощность составляла ~ 150 кВт, что создавало значительные проблемы с теплоотводом;

• оперативным средством связи пользователей являлся центральный пункт управления. Он содержал по 3 ряда тумблеров и индикаторов (по 43 неоновых лампочек каждый) и ряд индикаторов адреса выполняемой команды.

Несмотря на обилие радиоламп с ограниченным гарантийным сроком службы (до 500 часов) конструктивная реализация ЭВМ позволила довести среднее полезное время работы до 20 часов в сутки, но машина все равно с трудом справлялась с потоком задач. Из-за периодических сбоев при работе ЭВМ задачи считались с «двойным просчётом» и со сравнением контрольных сумм результатов.

Первыми программистами были выпускники ведущих вузов с физико-математической специализацией.

Они, в большинстве случаев, получив начальную постановку задачи и исходные данные, участвовали в её формализации, в разработке и отладке алгоритма и программы, зачастую выполняли функции операторов в организации вычислений.

БЭСМ-2 была создана С.А. Лебедевым в 1957 году как серийный аналог уникальной БЭСМ-1 и нашла применение в ряде НИИ СССР и за рубежом для численного решения широкого круга математических задач. Основные технические характеристики были аналогичны характеристикам БЭСМ, система команд машины отличалась тем, что были исключены редко использовавшиеся команды и добавлены некоторые новые

команды. ***Системное программное обеспечение в этих машинах отсутствовало.*** На серийных машинах БЭСМ-2 были решены тысячи задач: чисто теоретических, прикладной математики, инженерных. В частности, рассчитывались траектории полета первых космических аппаратов. Машина была разработана и внедрена в народное хозяйство коллективами ИТМ и ВТ АН СССР и завода им. Володарского (г. Ульяновск) в 1958-м году и производилась до 1962-го года.

В первую очередь, ЭВМ «Стрела» и БЭСМ-2 задействовали в военных целях – для изучения термоядерных реакций, расчета баллистических траекторий ракет и так далее. В 1956-м году Лебедев выступил с докладом на конференции в западногерманском городе Дармштадте. Академик устроил переполох, рассказав миру о том, что в СССР действует сверхбыстрая ЭВМ, – оказалось, что в Европе машине БЭСМ-1 не было равных.

Дальнейшее развитие вычислительных систем на протяжении нескольких лет было эволюционным. В 1958-м году на арену вышла система БЭСМ-2 с внешней памятью на основе ферритовых сердечников и увеличенным набором исходных команд. Впервые ЭВМ подготовили к серийному производству. Первые серьезные шаги по развитию централизованной производственной базы гражданских сфер применения ЭВМ были сделаны в конце 50-х годов после успешного завершения работ по созданию первых в нашей стране ***промышленных, универсальных вычислительных машин М-20*** (см. рис. 1). В 1958-м году в серию пошла машина М-20, созданная в коллективе С.А. Лебедева в ИТМ и ВТ (зам. главного конструктора М.К. Сулим и М.Р. Шура-Бура) [1, 3]. Скорость решения задач напрямую зависела от подготовленности программиста, – он должен был быстро реагировать на сбои, ошибки, отлично ориентироваться в переключателях пульта управления. Первые попытки реализовать системный язык программирования С.А. Лебедев предпринял еще при разработке М-20, машина понимала некоторые ***наглядные и интуитивные команды, мнемокоды.*** Это существенно расширило круг специалистов, способных взаимодействовать с ЭВМ.

Эта машина сыграла большую роль в развитии программирования, а позже на ее базе была создана транзисторная машина М-220. Создание машины М-20 являлось выдающимся достижением в развитии советской техники универсальных цифровых вычислительных машин. По своему быстродействию машина М-20 превосходила существовавшие отечественные и серийные зарубежные вычислительные машины. Благодаря большому быстродействию, совершенству логической

структуры и развитой системе оперативных и внешних запоминающих устройств, а также высокой надежности машины, она позволяла решать множество сложных задач, выдвигавшихся отраслями науки и техники.

Машина М-20 и ее аналог БЭСМ-4 имели следующие технические характеристики: быстродействие 20 тыс. операций в секунду, оперативная память на ферритовых сердечниках емкостью 4096 слов, представление чисел с плавающей запятой, разрядность 45, система элементов – ламповые и полупроводниковые схемы, внешняя память – магнитные барабаны и ленты, а также особенности:

• впервые в отечественной практике была применена автоматическая модификация адреса;

• совмещение работы арифметического устройства и выборки команд из памяти;

• введение буферной памяти для массивов, выдаваемых на печать, совмещение печати со счетом;

• использование накопителя на магнитной ленте с быстрым пуском и остановом;

• для М-20 разработана одна из первых технологических систем программного обеспечения ИС-2 (Институт прикладной математики АН СССР).

Вслед за М-20 были разработаны и освоены в серийном производстве машины «Урал-1», «Минск-1». Они вместе с их полупроводниковыми наследниками (М-220, Урал-11-14, Минск-22 и -32), созданными в 60-е годы, были основными в СССР, практически до освоения в серийном производстве машин третьего поколения, т. е. до начала 70 – х годов [1, 3]. Основную нагрузку по выпуску этих машин приняли на себя коллективы Московского завода САМ, Пензенского завода ВЭМ, а также вступившие в строй в 1959-м году Казанский завод ЭВМ, Минский завод математических машин, Астраханский завод «Прогресс» и ряд других предприятий. В эти же годы была существенно расширена научно-исследовательская и конструкторская база: в 1956-м году созданы НИИУВМ (Пенза) и НИИММ (Ереван); в 1958-м году – НИИ-250 (Пенза), а также ряд конструкторские бюро на заводах.

В середине 50-х годов в оборонных отраслях

промышленности и в организациях ***министерства обороны*** страны проявился интерес к применению цифровых вычислительных машин для решения задач ***обработки информации и управления в системах военного назначения.*** Начались активные, секретные работы по освоению применения цифровой вычислительной техники для систем противовоздушной и противоракетной обороны, для контроля космического пространства и управления полетом в авиации и в космосе, для управления войсками и средствами вооружения разных видов. Многие из этих задач ***принципиально отличались по своему характеру и масштабу*** от ставших к тому времени традиционными ***вычислительных задач в гражданских областях и в науке.*** В них преобладали: логические операции, большая размерность, реальный масштаб времени и ряд других специфических свойств и требований. Очень быстро увеличивались номенклатура и объем функций систем, которые требовалось автоматизировать. Для реализации таких функций были необходимы значительные ресурсы памяти и производительности ЭВМ, а также большие коллективы специалистов, способные создавать крупные комплексы алгоритмов и программ в допустимые сроки. Уже первые комплексы программ военного назначения в 50-е годы достигали нескольких десятков тысяч команд, для чего было необходимо разрабатывать и применять некоторые ***методы программной инженерии.*** В результате начало активно развиваться ***специфическое направление вы числительной техники и программирования для крупных систем реального времени оборонного назначения*** [3, 9].

Это направление почти одновременно начало формироваться в оборонных отраслях промышленности и на предприятиях в нескольких проблемноориентированных областях: для сухопутных, авиационных, морских, ракетных и других систем. Для последующего развития вычислительной техники существенными оказались особые ***требования заказчиков*** различных областей применения. В результате ЭВМ разделились на два класса: на ***стационарные***, работающие в помещениях, и на ***мобильные,*** размещаемые на подвижных (транспортабельных) или движущихся (бортовых) объектах (в том числе, необслуживаемых). Эти факторы определили ***большие***

принципиальные различия в архитектуре, технических, климатических и массогабаритных характеристиках этих двух классов, специализированных ЭВМ оборонного назначения, а также в программировании для них. Первый класс тяготел к архитектурам и конструктивам стационарных, универсальных ЭВМ с необходимыми расширениями и модификациями для специализированного применения. Машины второго класса – мобильные, отличались наибольшей спецификой свойств задач и характеристик внешней среды применения, от остальных типов ЭВМ.

Начали разрабатываться относительно небольшие, бортовые, мобильные, а также крупные ***территориально-распрелеленные вычислительные системы*** на базе средств телекоммуникации, функционирующие в реальном времени. Все эти работы проводились ***в режиме строгой секретности,*** и каждая функционально законченная оборонная система создавалась практически независимо как от достижений за рубежом, так и от методов и результатов на других отечественных предприятиях. На них акцентируется последующее изложение в главе 2.

Для исследований, моделирования и постановки задач систем вооружения для оборонной промышленности в 1954 – м году был создан ВЦ-1 министерства обороны СССР – ***первый в стране профильный вычислительный центр.*** В научно-производственном аспекте по широте научных исследований и количеству разработчиков и специалистов в 1950-е годы это был самый мощный вычислительный центр в Советском Союзе и один из самых мощных в мире [11]. ВЦ-1 МО (впоследствии ЦНИИ-27 Министерства обороны СССР) был создан по инициативе Анатолия Ивановича Котова, который он и возглавил. ВЦ-1 стал одним из ведущих оборонных научных центров страны. В 1952-м году А.И. Китов защищает первую в СССР кандидатскую диссертацию, посвященную вопросам программирования. Название этой диссертации – «Программирование задач внешней баллистики ракет дальнего действия». В этом же году он организовал и возглавил первый в стране отдел вычислительных машин в Академии артиллерийских наук. После упразднения этой Академии в 1953-м году отдел А.И. Китова вместе с его начальником был переведён в подчинение Артиллерийской военно-инженерной академии им. Ф.Э. Дзержинского, а затем он возглавил ВЦ-1.

Осенью 1959-го года Анатолий Иванович послал в ЦК КПСС на имя Н.С. Хрущёва разработанный им проект создания общегосударственной автоматизированной системы управления для вооруженных сил и для народного хозяйства страны на базе

Единой государственной сети вычислительных центров (ЕГСВЦ) – так называемый проект «Красная книга». В преамбуле этого доклада давалась резкая критика текущего состояния дел в стране с внедрением ЭВМ, и в первую очередь, в министерстве обороны СССР. Это предопределило негативное отношение к докладу А.И. Китова партийного и военного руководства СССР. Главное заключалось в том, что работники аппарата ЦК КПСС и, в частности, аппарата МО СССР почувствовали, что коренная перестройка управления на основе этого проекта приведет к устранению многих из них от рычагов государственной власти. В результате, А.И. Китов был исключен из КПСС и снят с престижной генеральской должности, которую он занимал в ВЦ 1 [11].

В 1965–1972 гг. А.И. Китов был Главным конструктором Отраслевой автоматизированной системы управления (ОАСУ) министерства радиопромышленности СССР и директором Главного вычислительного центра этого министерства. Потом около десяти лет он работал Главным конструктором АСУ «Здравоохранение». Опубликовал ряд основополагающих монографий и статей по вопросам применения ЭВМ и экономико-математических методов в области экономической информатики и медицинской информатики, в 1968-м году защитил докторскую диссертацию по гражданской тематике.

После ВЦ-1 в 50-е годы министерством обороны были организованы ВЦ-3 для разработки и исследования вычислительных систем и программирования авиационных систем, а также ВЦ-4 для артиллерийских и ракетных систем.

Отсутствие в стране в 50-е годы развитой централизованной промышленности электронных компонентов для ЭВМ являлось причиной их разработки зачастую теми же предприятиями, которые создавали архитектуру специализированных ЭВМ и системы управления в целом. Вследствие этого ***элементная база машин часто была полукустарной, малотиражной и разнотипной,*** не отличалась высоким качеством и технологическим уровнем. Необходимость для многих предприятий вести разработку систем по полному циклу, начиная с создания элементной базы ЭВМ и далее всей вычислительной техники и программных средств, не только приводила к множеству параллельных, не унифицированных разработок, но и значительно увеличивала длительность и стоимость проектов систем. Впоследствии это отразилось на сложности производства множества разнотипных ЭВМ, на трудностях сопровождения и модернизации систем в целом. С.А. Лебедев был убежден, что в разработках ЭВМ должна использоваться отечественная

элементно-конструкторская база. ИТМ и ВТ был первым заказчиком дискретных интегральных и больших интегральных схем в министерстве электронной промышленности СССР. Отставание в технологии компенсировалось передовыми схемотехническими и архитектурными решениями. БЭСМ-6 была одной из лучших в мире ЭВМ по архитектурным и схемотехническим решениям (см. главу 2).

В 1960-м году были начаты работы по созданию ***семейства*** полупроводниковых ЭВМ «Урал» [3]. Основные черты нового поколения машин были сформулированы Баширом Искандеровичем Рамеевым в 1959-м году. В соответствии с ними были определены: состав семейства машин, их структура, архитектура, интерфейсы, принципы унификации, утверждены технические задания на устройства, ограничения на типы используемых комплектующих изделий. В конце 1962 года была закончена разработка унифицированного комплекса полупроводниковых элементов «Урал-10», рассчитанного на автоматизированное производство. Хотя элементы разрабатывались для использования в серии ЭВМ «Урал-11» – «Урал-16», они нашли широкое применение и в других средствах вычислительной техники и автоматики. В результате принятых правительством мер к началу 60-х годов были практически завершены все работы, связанные с созданием и освоением серийного производства полупроводниковых ЭВМ [3]. Это позволило прекратить, начиная с 1964 года, производство ламповых машин первого поколения и с 1965 года начать массовое ***производство полупроводниковых машин*** Урал-11 – Урал-14; Минск-22; Минск-23; БЭСМ-4; М-220; Раздан-3 и др.

В 1965 – 66-е годы все предприятия, НИИ и КБ в области вычислительной техники были переданы ***в состав двух министерств —*** министерства радиопромышленности (универсальные и специальные, бортовые, военные ЭВМ) и министерства приборостроения, средств автоматизации и систем управления (промышленные управляющие ЭВМ) [3, 10]. Работа предприятий в этих условиях совпала с началом активного создания и подготовки производства ЭВМ третьего поколения (на интегральных схемах). Трудности этого периода были связаны не только с решением научно-технических и технологических проблем (от архитектуры до элементной базы новых ЭВМ). Необходимо было решать большое количество сложных проблем создания в стране практически с нуля ***крупной отрасли вычислительной техники,*** базирующейся на новой технологии и широкой номенклатуре ранее не выпускавшихся средств, с переходом на внутриотраслевую специализацию. Освоение новых

изделий во многих случаях шло одновременно со строительством самих заводов и обучением специалистов и сопровождалось множеством других проблем.

Решать все эти проблемы необходимо было ***в крайне ограниченное время*** (3–5 лет) с одновременным увеличением выпуска ЭВМ более чем в три раза при существенном увеличении состава оборудования в каждой машине. Реализовать эту задачу предполагалось в дальнейшем за счет разработки и освоения в серийном производстве нескольких типов программно совместимых вычислительных машин, построенных на единой конструктивно-технологи-ческой базе. Увеличение объемов производства достигалось за счет специализации производства и его лучшего технологического оснащения. Сокращение сроков разработки предусматривалось как за счет использования ***(легального)*** опыта ведущих западных фирм, на основе заключенных с ними соглашений, так и за счет привлечения к разработке и производству новых ЭВМ коллективов практически всех предприятий и организаций, ранее работавших над созданием ***множества разнообразных, собственных*** ЭВМ. Реализации этих задач было посвящено ***постановление правительства*** 1967-го года, в котором были сформулированы задачи и предусмотрены необходимые меры для обеспечения их выполнения материальными, производственными и финансовыми ресурсами [3, 11].

Это ***постановление определило создание в стране отрасли вычислительной техники,*** т. к. оно охватывало решение всех основных проблем – от разработки и освоения производства материалов и элементной базы до обеспечения производства нового поколения ЭВМ и повышения эффективности его использования в народном хозяйстве. ***Постановлением было предусмотрено***:

• увеличение мощностей по производству средств вычислительной техники с 304 млн. рублей в

1965-м году до 1000 млн. рублей в 1970-м году и до 3000 млн. в 1975-м году;

• рост выпуска средств вычислительной техники с 2470 млн. рублей в 1966 – 1970-е годы до 7500 млн. рублей в 1971 – 1975-е годы;

• увеличение выпуска ЭВМ с 5800 машин в 1966 – 70-е годы до 20000 машин в 1971 – 1975-е годы.

Только по министерству радиопромышленности СССР постановлением было предусмотрено строительство 14 новых заводов и реконструкция 11 существующих. Аналогичное развитие было предусмотрено и по предприятиям министерства

приборостроения, средств автоматизации и систем управления и министерства электронной промышленности. Кардинальные решения были приняты по развитию мощностей по производству элементной базы машин третьего поколения, практически «с нуля» до 65 млн. интегральных схем в год. Эта программа максимум не была полностью выполнена, но она способствовала тому, что в стране примерно вдвое выросли производственные мощности по выпуску компонентов и систем вычислительной техники. В результате ***в*** 60-е годы ***были созданы предпосылки*** для последующей разработки таких высокопроизводительных систем, как БЭСМ-6, 5Э26, АС-6, МВК Эльбрус, М-13 (см. главу 2).

1.2. Начало истории отечественного программирования в 1950-е – 60-е годы

Первые программы определялись в ЭВМ ***установкой ключевых переключателей*** на передней панели вычислительного устройства. Очевидно, таким способом можно было составить только очень небольшие программы. С развитием вычислительной техники ***появился машинный язык,*** с помощью которого программист мог задавать команды, оперируя с ячейками памяти, полностью используя возможности машины. «Слова» на машинном языке, представляло собой одно элементарное действие для центрального процессора, такое, например, как считывание информации из ячейки памяти. Каждая модель процессора имела свой собственный набор машинных команд, хотя большинство из них совпадало. Тогда еще не было компиляторов и приходилось все писать числами. Это был адский труд – постоянно держать в памяти таблицу машинных кодов и вводить их в ЭВМ.

Со временем ЭВМ стала умнеть, но самое главное, она все также оперировала двоичными числами, однако делала это намного быстрее. Программист – это человек, и ему очень тяжело создавать логику в числах. Намного легче работать с привычными словами. В случае, когда нужно иметь эффективную программу, вместо машинных языков начали использоваться близкие к ним ***машинно-ориентированные языки – ассемблеры.*** Использовались мнемонические команды взамен машинных команд. Но даже работа с ассемблером достаточно сложна и требует специальной подготовки.

На протяжении 60-х годов запросы на разработку программного обеспечения быстро возросли и программы становились очень большими. Руководители начали понимать, что создание программного обеспечения – гораздо более сложная задача, чем они себе представляли. Это привело к тому, что было разработано ***структурное – модульное программирование.*** С развитием структурного программирования следующим достижением были процедуры и функции. Если задача выполняется несколько раз, то ее можно объявить, как функцию или процедуру и в выполнении программы просто вызывать ее. Общий код программы в данном случае становится меньше. ***Функции*** позволяют создавать модульные программы, в основе которых лежит представление программы в виде иерархической структуры блоков. ***Класс —*** это структура, которая имеет свои

переменные и функции, которые работают с этими переменными. Это было очень большое достижение в области программирования. Программирование можно было разбить на классы и тестировать не всю программу, состоящую из строк кода, а разбить программу на группу классов, и тестировать каждый класс. Это существенно облегчило написание программного продукта.

Следующий шаг был сделан в 1954-м году, когда на Западе был создан ***первый язык программирования высокого уровня – Фортран.*** Языки высокого уровня имитируют естественные языки, используя некоторые слова разговорного языка и общепринятые математические символы. Эти языки более удобны для человека, с помощью них можно писать программы до нескольких тысяч строк длиной. Однако легко понимаемый в коротких программах, этот язык становился нечитаемым и трудно управляемым, когда дело касалось больших программ. Решение этой проблемы пришло после изобретения на Западе языков структурного программирования, таких как ***Алгол*** (1958), ***Паскаль*** (1970), ***Си*** (1972). С этого момента начался языковый бум. Языки программирования стали появляться один за другим. Так появились C+, ADA, FoxPro, Basic, Pascal и др. На сегодняшний день существует тысячи языков программирования. Из них популярность и известность получили только некоторые. Они отличаются простотой, быстротой, гибкостью и другими свойствами, удобными в некоторой определенной области использования.

Споры программистов перенеслись в другую плоскость – ***какой язык лучше.*** Большее предпочтение отдавалось универсальным языкам программирования, способным предоставлять эффективный инструментарий для решения разнообразных вычислительных задач. Все современные реализации широко распространенных языков обладают сходными характеристиками: начиная от скорости написания программ и кончая производительностью полученного кода.

Теперь приверженцы языка ассемблер утверждали, что их код самый быстрый, а любители языков высокого уровня утверждали, что они напишут программу быстрей, чем самый лучший программист на языке ассемблер. Однако программы на языках высокого уровня занимали больший объем памяти и работали медленнее. В вычислительных задачах победила в основном скорость разработки и «удобство» языка программирования для определенного класса задач.

Ассемблер не ушел из практики, когда необходимо было создавать эффективные по объему программы реального времени

высокой производительности и занимающие минимальный объем памяти (см. главу 4). В 60-е – 70-е годы большинство программных продуктов ***в оборонной промышленности для мобильных и бортовых ЭВМ создавались на ассемблерах,*** адаптированных на архитектуру соответствующих машин. Только в 80-е годы в связи с ростом ресурсов специализированных машин начали использоваться алгоритмические языки высокого уровня, несмотря на возможное расширение программ при трансляции, по сравнению с программами, разработанными на ассемблере.

С середины 50-х годов во всех странах, производивших вычислительную технику, начался бурный ***процесс «языкотворчества» —*** создание нескольких сотен проблемно-ориентированных языков [1, 4]. Многим программистам казалось, что небольшие улучшения языков программирования способны радикально повлиять на разработку и качество программ, на использование скудных ресурсов ЭВМ. Однако это требовало создания соответствующих трансляторов и средств отладки, вследствие чего энтузиазм языкотворчества постепенно угас. Сложной задачей для системных программистов того времени было создание трансляторов для конкретных типов машин. Создание каждого транслятора с машинно-независимого языка программирования считалось крупным научным и практическим достижением. Большое число различных типов машин и различных языков требовало трудоемкой работы высококвалифицированных программистов и математиков по разработке трансляторов. Соответственно, возникла необходимость создания ***небольшого числа стандартизированных языков*** и программно-преемственных ***семейств*** вычислительных машин. Это потребовало глубоких теоретических исследование в теории алгоритмов, схем программ, теории формальных грамматик.

В СССР в 60-е годы был создан алгоритмический ***язык РЕФАЛ,*** в основе которого лежала теоретическая модель процесса, реализуемого нормальными алгоритмами Маркова. Его использование в нашей стране позволило создать ряд оригинальных программных продуктов, не имевших аналогов за рубежом. Однако РЕФАЛ испытал судьбу многих отечественных находок. Сходная судьба была у языков программирования ***семейства Аналитик,*** созданных в Институте кибернетики АН УССР для ЭВМ серии «МИР». Эти машины, по существу, были первыми персональными ЭВМ (к сожалению, тогдашняя элементная база не позволила свести их габариты к настольным). Однако, несмотря на передовые принципы, заложенные в структуру и функции языков семейства Аналитик, они также не

стали достоянием мирового сообщества программистов, хотя иностранные эксперты достаточно высоко оценивали достижения программирования в СССР [4].

Исследования в области параллельного программирования в СССР начались в середине 60-х годов, когда в Институте математики СО АН СССР (Новосибирск) и в Московском энергетическом институте возникли первые коллективы, заинтересовавшиеся теорией параллельных процессов в вычислительных системах, состоящих из однородных или неоднородных машин [4]. Первые монографии по теории вычислительных систем и параллельных вычислений вышли в нашей стране с большим опережением аналогичных изданий за рубежом. Отечественные специалисты первыми в мировой науке дали постановку и предложили решения таких задач, как сегментация алгоритмов и программ, планирование выполнения больших программ на вычислительных системах, динамическое диспетчерование потока программ и сегментов программ, асинхронная организация протекания процессов. В это время было предложено несколько оригинальных моделей для параллельных вычислений, заново открытых потом в США и других странах.

В 1953-х годах Л.В. Канторович разработал ***технологию крупноблочного программирования,*** которая давала обозримое описание крупных программ и обеспечивала формализацию, достаточную для исследования синтаксических структур программ и создания программирующих программ [4]. Идеи, высказанные в этих работах, ***предшествовали развитию программной инженерии.*** Работа школы протекала весьма активно в 1950 – 60-е годы. Характерной особенностью крупноблочных систем являлось то, что они оперировали не с индивидуальными числами и символами, а с величинами – укрупненными агрегированными информационными объектами. Такие укрупненные структуры данных (матрицы, векторы, последовательности, деревья, схемы и т. д.) выступали как целое в вычислительных планах; стандартные способы обработки отдельных компонентов выполнялись автоматически на нижних уровнях. Это вносило ***иерархическую структуру в языки программирования,*** освобождая верхние уровни от ненужной детализации. Существенно, что вычислительный процесс мыслился при этом также «объемным», протекающим одновременно, либо попеременно на каждом из этих уровней. Громоздкие и трудоемкие вычисления часто чрезвычайно упрощались при переходе на другой уровень. Представлялось, что в разумной стратегии переходов с одного уровня на другой

кроется значительный резерв для повышения экономики вычислений. На этой же уровневой основе была создана оригинальная теория и методология трансляции, гибко сочетающая компиляцию и интерпретацию.

Ряд объективных обстоятельств способствовал тому, что до середины 60-х годов ***программирование в СССР развивалось до некоторой степени автономно от зарубежного*** [1, 3, 4]. К этим обстоятельствам относились:

• более позднее начало массового производства электронной вычислительной техники (примерный сдвиг – 5 лет);

• меньшее количество доступных вычислительных ресурсов, приведшее к не столь широкому размаху работ, как в США или в Англии;

• практическое отсутствие импорта вычислительных машин и технологий;

• языковый барьер и сравнительно менее интенсивные личные контакты специалистов (в частности, вследствие секретности);

• некоторые общие отличия в организации и стиле научных исследований и производства.

Заметное влияние на общее развитие программирования в мире оказали работы Ю.И. Янова, приведшие к созданию теории схем программ, и некоторые работы по оптимизации трансляции. Существенный вклад в мировую тенденцию внесло широкое распространение Алгола-60 в СССР. Большая часть результатов представляла независимые, индивидуальные, практические разработки, без которых невозможно было полноценное развитие отечественного программирования.

Одним из факторов, сузившим фронт работ по автоматизации программирования в несекретных отраслях народного хозяйства, в это время было ***преобладание научных применений универсальных ЭВМ.*** Большая часть программистов в СССР была математиками с университетским образованием. Почти все успешные ***«художественные»*** экспериментальные и вычислительные программы кое-как переделывались в программный продукт, и это «кое-как» иногда мешало эксперименту и не давало должного эффекта при применении продукта. При всех положительных сторонах этого обстоятельства потребовалось длительное время, пока была осознана и реализована необходимость сбалансировать эту сторону вузовского образования с ***воспитанием способности к инженерному стилю*** работы, столь необходимому в системном программировании ***для создания крупных программных продуктов реального времени*** (см. главу 4).

В ***середине*** 50-х годов появились небольшие по численности

группы математиков, привлеченных к разработкам проектов вычислительных машин, проводившихся в небольшом числе проектных организаций и институтов в Москве, Ленинграде, Киеве, Минске, Пензе. Каждая вновь разрабатываемая машина, прежде всего, требовала создания для нее операционной системы и программ вычисления элементарных функций. При этом необходимо было добиваться предельной эффективности таких вычисления на данной конкретной архитектуре и ресурсах ЭВМ. Это требовало от математиков высокого уровня понимания деталей логики работы процессора. Возможно, что именно это явилось отличительной чертой отечественных школ программирования, чертой теснейшей их связи с инженерными разработками, которая определила в дальнейшем, как достоинства, так и недостатки в работе этих школ.

Теоретические исследования методов программирования для ЭВМ в 1950-е – 60-е годы, активно проводились в Московском, Ленинградском и Киевском университетах, в Институте автоматики и телемеханики АН СССР, в Вычислительном центре АН СССР [1, 4]. В 1950 году в ИТМ и ВТ начал работать первый постоянный семинар по программированию, которым руководил Л.А. Люстерник, в МГУ в 1952-м году была основана кафедра вычислительной математики (ее возглавил С.Л. Соболев). В 1953-м году в Математическом институте АН СССР был создан отдел программирования во главе с А.А. Ляпуновым, а в 1955-м году был основан Вычислительный центр МГУ, специализировавшийся на разработке и применении ***вычислительных методов для решения научных и прикладных задач.***

В 1952-м – 53-м годах А.А. Ляпуновым был предложен операторный метод для описания программ. Практически впервые был создан способ представления программ на обозримом уровне. Вместо неэффективного для человека написания программ в машинных кодах было предложено формализованное представление программ на языке высокого уровня. Особенно важным было то, что операторный метод позволил подготовить теорию синтаксических структур программ. В 1953-м году А.А. Ляпунов сформулировал постановку ***задачи автоматизации программирования.*** Эта оригинальная постановка была успешно использована в первых отечественных трансляторах, называвшихся тогда ***программирующими программами*** (ПП). Летом 1954-го года появилась программирующая программа ПП-1, разработанная в отделе прикладной математики Института математики АН СССР, а в 1955-м году – ее улучшенный вариант ПП-2. К середине 50 – х

годов у ведущих специалистов в области вычислительной техники сформировалось представление о путях развития отечественной информатики и программирования.

В середине 1957-го года, Виктор Михайлович Глушков, определил направления стратегических исследований в области информатики [3]. По его мнению, основой прогресса ***развития*** вычислительных машин должна была стать теория их работы, разработка методов автоматизации проектирования ЭВМ и ***автоматизации программирования.*** Он подчеркивал важную роль исследований в области теории алгоритмов и теории конечных детерминированных и стохастических автоматов, принципиальное значение разработки методов символьных преобразований на ЭВМ. Отмечалась центральная роль, которую играет ***задача оптимизации размера при трансляции программ*** (особенно для управляющих машин), а также указывалось на обратное влияние развития вычислительных машин на дальнейшие работы в области вычислительной математики. В конце 1959-го года в Москве, в МГУ состоялось «Всесоюзное совещание по вычислительной математике и вычислительной технике». Это было большое научное собрание с почти 2000 участников и 217 докладами, прочитанными на четырех секциях.

Постепенно складывалась ***концепция системного математического (программного) обеспечения*** ЭВМ – интегрированной и удобной в работе системы ***различных средств автоматизации программирования*** (библиотеки, трансляторы, средства отладки), сопряженных с определенной дисциплиной реализации задач на машине [5]. Проблеме математического обеспечения ЭВМ было много препятствий научно-технического и организационного характера, одно из которых – ***слабая разработанность концепции программного продукта и его производства.*** Для М-20 в то время таким средством для вычислений была библиотека стандартных подпрограмм. Идея превращения библиотеки в переносимый и общий программный продукт стала для М.Р. Шура-Буры на некоторое время главной задачей, на решении которой со временем сформировались более общие взгляды на системное программное обеспечение. Необходимо было найти некоторый объект конструирования и научной работы, который одновременно решал бы задачу унификации математического обеспечения, мог бы быть эффективным средством автоматизации программирования и выдвигал бы новую научную проблематику. Таким ***корневым объектом в стране стал Алгол-60.***

В начале 60-х годов был опубликован алгоритмический язык Алгол-60, рекомендованный в качестве международного

стандарта для публикаций вычислительных алгоритмов. В нашей стране ***Алгол-60 был принят в качестве государственного стандарта.*** Использование других языков для программирования вычислительных задач не рекомендовалось и это касалось даже широко применявшегося за рубежом более простого языка Фортран. После этого в стране началась активная разработка трансляторов с Алгола– 60 для нескольких типов машин. Попытки создать транслятор с полного языка Алгол-60 за рубежом не удались. В ИПМ такой транслятор был создан для машины «Стрела», а затем для машины М-20, что явилось достижением мирового уровня.

В то же время стала совершенно очевидной общенаучная ценность этого документа, которая требовала его широкого распространения. Ситуация весной 1960-го года оказалась весьма благоприятной для принятия Алгола-60 в качестве единого языка программирования научных и инженерных применений ЭВМ. В июне в Вычислительном центре АН СССР ***координационное совещание по вопросам реализации Алгола-60.*** На фоне общей и во многом разнонаправленной активности выделились три проекта реализации языка для М-20, получившие, соответственно, названия ТА-1, ТА-2 и Альфа. Начавшись как три независимые и подчас конкурирующие разработки, они в процессе развития приобрели взаимодополняющие свойства, решив удовлетворительно проблему снабжения М-20 трансляторами с Алгола-60.

В декабре 1960-го года в МГУ состоялась ***рабочая конференция «Построение программирующих программ*** на основе языка Алгол-60» [1, 4]. К этому времени у разработчиков уже сложились общие подходы к реализации языка и выбору схем трансляции. В ТА-1 благодаря отказу от возможной рекурсивности процедур и ряду других ограничений, была выбрана компактная и быстрая схема трансляции без оптимизации. Главной задачей ТА-2 стала реализация практически полного языка без существенной потери в качестве реализации. В разработке системы Альфа (Андрей Петрович Ершов) было поставлено в качестве главной цели обеспечение высокого качества (по объему рабочих программ) с сохранением приемлемой скорости трансляции. Употребление универсального языка программирования снимало задачу перевода программ с одной машины на другую, дало возможность сокращать дублирование работ по составлению программ для различных типов машин и существенно облегчило обмен информацией между отдельными группами специалистов, работающих в области программирования. В докладе разработчиков системы

Альфа был показан ***классический прецедент просчета в определении плановых экономических показателей и трудоемкости больших программных проектов*** с универсальным коэффициентом недооценки трудоемкости в 2–3 раза, подтвержденном впоследствии многими проектами. Авторы клали на разработку системы 15 человеко-лет для построения 15000 команд, затратив на самом деле свыше 30 человеко-лет и соорудив систему в 45000 команд (см. главу 5).

Библиотека программ ИС-2, трансляторы ТА-1 и ТА-2 стали ***первыми образцами программных продуктов,*** которые поставлялись вместе с оборудованием машин заводом-изготовителем, образуя интегрированную систему поддержки программирования. В трансляторе Альфа были систематически учтены очень ограниченные ресурсы технологической ЭВМ:

• функция расстановки, для ускорения работы транслятора, в частности, для экономии совпадающих выражений;

• многовариантная система программирования процедур и циклов, основанная на анализе структуры программы;

• реализована глобальная экономия памяти;

• осуществлен ряд оптимизационных преобразований на уровне промежуточного языка, в частности, объединение циклов с одинаковыми заголовками и чистка циклов.

После первых успехов в области создания трансляторов ТА-1, ТА-2 и Альфа в 1964-м – 65-м годах ***отечественные исследования в области автоматизации программирования*** продолжали сохранять высокий темп развития. Появление ЭВМ 2-го поколения (Минск 2, БЭСМ-4, М-220 и др.) в целом определило созревание концепции математического (программного) обеспечения и ***идентификацию системного программирования.*** Интенсивная работа над трансляторами с Алгола-60 привела к практическому исчезновению профессии вспомогательного программиста-кодировщика машинных кодов и замене ее на профессионального системного программиста. Все первые трансляторы писались в восьмеричном машинном коде с минимальными средствами автоматизации. Это привело к появлению первых языков системного программирования и к первой системе построения трансляторов, основанной на промежуточном универсальном машинно-ориентированном языке АЛМО. Пионерскими в области программирования были работы А.П. Ершова [1, 4] по компиляции с минимальной памятью и по теории программирования (схемы Янова – Ершова).

Параллельно работам по Алголу-60 развивались события, приведшие к организации ***ассоциации пользователей ЭВМ М-20.*** В середине 1961-го года решением Президиума Академии наук

СССР ассоциация получила статут юридического лица и официальное название «Комиссия по эксплуатации вычислительных машин М-20». Деятельность Комиссии была важна не только созданием прецедента, за которым последовало создание аналогичных ассоциаций для БЭСМ-2, для семейства «Урал», а также для серии «Минск», но и ускорением разработки ***концепции математического (программного) обеспечения.***

В 1964-м году началось проектирование первых программных операционных систем для пакетной обработки с использованием загрузчиков и трансляторов ассемблеров, работающих в автоматическом режиме с помощью языков управления заданиями. Большую роль в формировании современного взгляда на математическое обеспечение и архитектуру ЭВМ сыграл ***Конгресс ИФИП*** 1965-го года, когда концепции совместимых семейств машин, разделения времени, мини-ЭВМ стали объектом делового интереса советских специалистов. Начиная с 1964-го года, разработка математического (программного) обеспечения стала ***элементом государственной технической политики.*** Государственный комитет по науке и технике (ГКНТ) был назначен координатором работ по математическому обеспечению существующих машин и генеральным заказчиком для промышленности на математическое обеспечение вновь создаваемых ЭВМ. Ассоциации пользователей ЭВМ активно представляли научно-техническое общественное мнение и играли существенную роль в распространении новых программ. Апробация новых систем программирования, а впоследствии и операционных систем проводилась междуведомственной комиссией по математическому обеспечению под председательством академика А.А. Дородницына и целевыми комиссиями, осуществлявшими приемку новых компонентов математического обеспечения.

Первая докторская диссертация по программированию была защищена в Киеве Е.Л. Ющенко в 1966-м году по материалу разработки серии трансляторов на основе адресного языка. Л.Н. Королев стал в 1965-м году первым профессором – программистом [4]. Первое время все эти диссертации причислялись к существовавшим в то время родственным специальностям: вычислительной математике, счетно-решающим устройствам и т. д. В середине 60-х годов под влиянием серии работ по теоретическому программированию была образована ***новая специальность «математическая логика и программирование».*** А.П. Ершов был первым программистом, ставшим членом-корреспондентом АН СССР в 1970-м году, а в 1980-м году – ее действительным членом. Сознавая социальные

последствия использования ЭВМ и культурное значение программирования (он называл его ***«второй грамотностью»***), А.П. Ершов активно проповедовал его введение в школьную информатику и в курс «Основы вычислительной техники и обработки информации». Его желание компьютеризировать школу преследовало двойную цель: развить в молодых людях интеллектуальный дар программирования и обогатить их мощью информационной обработки.

А.П. Ершов активно поддерживал международный научный обмен и сотрудничество. Он постоянно участвовал в различных комитетах и конференциях IFIP, организовывал многочисленные международные конференции в Новосибирске и других регионах Советского Союза. А.П. Ершов инициировал (и часто редактировал) переводы западных книг по информатике. Установление и развитие личных и профессиональных связей между иностранными учеными и их советскими коллегами было целью, которой он посвятил значительную часть своей энергии.

Взгляды АП. Ершова на программирование, выраженные в серии очерков, начатой в 1972-м году, привлекли широкое внимание во всем мире. Описывая свою профессию поэтически, он утверждал [6]:

«…Программист должен обладать способностью первоклассного математика к абстракции и логическому мышлению в сочетании с эдиссоновским талантом сооружать все что угодно из нуля и единицы. Он должен сочетать аккуратность бухгалтера с проницательностью разведчика, фантазию автора детективных романов с трезвой практичностью экономиста. А, кроме того, программист должен иметь вкус к коллективной работе, понимать интересы пользователя и многое другое.

Машина, снабженная программой, ведет себя разумно. В этот кульминационный момент программист, по существу, представляет троицу. Он ощущает себя отцом – как создатель программы, сыном – как брат машины, выполняющей программу, и носителем святого духа – как тот, кто вложил жизнь в сочетание программы и машины».

Сильным тормозом в развитии и внедрении автоматизации программирования в 50-е годы было отсутствие буквенно-цифровых устройств ввода-вывода, которые стали общедоступными только с машинами 2-го поколения. Это затрудняло внедрение и ослабляло потребность в разработке комфортных средств отладки. Сужалась и даже становилась в значительной степени бесполезной методика символического кодирования и программирования. Более глубоким последствием стал недостаток внимания к текстовому представлению входных

программ для первых трансляторов. Другим примером ограничительного влияния оборудования являлось довлевшее над большинством разработчиков трансляторов требование воссоздать средствами входных языков «любую» машинную программу.

1.3. Первые комплексы программ для оборонных систем в 1950-е – 60-е годы

Большое число крупных оригинальных, специализированных исследований и производственных работ по программированию, которые были ***связаны с созданием оборонной техники*** для авиации, космических, ракетных, морских и наземных систем, долгое время оставались секретными. Они охватывали множество сложнейших вычислительных задач, а также специфические задачи управления и обработки информации в динамических оборонных системах реального времени. Этот класс задач был не актуальным и недоступным для индивидуальных программистов в вузах и научных учреждениях, практически не упоминался в открытой печати, однако в 60-е годы и в дальнейшем ими были заняты в стране сотни тысяч специалистов, связанных с созданием сложных программ для ЭВМ.

Сергей Алексеевич Лебедев являлся ***инициатором внедрения электронной вы числительной техники в оборонные системы***: в радиолокацию, ракетостроение и системы передачи данных. По его инициативе впервые в СССР, а возможно и в мире, проведены работы по фиксированию данных с радиолокационных станций сопровождения целей в цифровом виде и по передаче управляющей информации для наведения самолета или ракеты на цель. Преимущества вычислительной техники в системах военного применения были впервые продемонстрированы под руководством С.А. Лебедева в «Системе А» – ***экспериментальной системе противоракетной обороны (ПРО).*** Данный комплекс управлял радиолокационной станцией дальнего обнаружения и сопровождения цели и точного наведения противоракеты на баллистическую ракету противника [8, 9]. Были разработаны принципы построения вычислительных средств противоракетной обороны и создан высокопроизводительный вычислительный комплекс для решения задач высококачественного автоматического управления сложными, разнесенными в пространстве объектами, работающими в реальном масштабе времени. В его состав входили ЭВМ М-40, радиолокаторы обнаружения и сопровождения цели, радиорелейные линии передачи данных в замкнутой системе точного наведения ракеты, система контрольно-измерительной аппаратуры. ЭВМ М-40 начала выполнять сложные боевые задачи в 1957-м году. Впервые были предложены принципы распараллеливания вычислительного процесса за счет аппаратных средств. В марте

1961 года на этом комплексе впервые в мире была ликвидирована боевая часть баллистической ракеты осколочным зарядом противоракеты. За эти работы коллектив ведущих разработчиков комплекса, в том числе С.А. Лебедев и В.С. Бурцев, был удостоен Ленинской премии. ЭВМ М-50, введенная в строй в 1959 году, и явилась модификацией ЭВМ М-40, обеспечивающей выполнение операций с плавающей запятой и рассчитанной на применение в качестве универсальной ЭВМ. На базе М-40 и М-50 был создан двухмашинный комплекс.

С 1953-го года Михаил Романович Шура-Бура работал в Отделении прикладной математики Математического института им. В.А. Стеклова, созданном М.В. Келдышем в 1953-м году и преобразованном затем в Институт прикладной математики АН СССР (ныне ИПМ РАН им. М.В. Келдыша). В эти годы главной задачей и организационным успехом М.Р. Шуры-Буры как руководителя отдела программирования в ИПМ было формирование отдела [11]. Первым результатом работы отдела в 1953-м – 1955-м годах было применение программ ***для расчета энергии взрывов*** при моделировании ядерного оружия на ЭВМ «Стрела». Постановки задач и методы расчетов для этих программ готовили отделы математиков (А.Н. Тихонов, А.А. Самарский, И.М. Гельфанд). Программирование задач такой сложности в машинных кодах на ЭВМ, имевшей оперативную память емкостью всего 1000 ячеек, неработающий накопитель на магнитной ленте и частые сбои в арифметике и управлении, требовало от программистов виртуозного умения и оригинальных находок в организации отладки программ и счета. В 1954 – м году М.Р. Шура-Бура защитил диссертацию на соискание ученой степени доктора физико-математических наук.

В середине 50-х годов отдел программирования был привлечен М.В. Келдышем к ***расчетам траекторий искусственных спутников Земли*** (ИСЗ). Программы, разработанные сотрудниками отдела, возглавляемого Михаилом Романовичем, для ЭВМ «Стрела», а затем М-20, должны были обеспечивать круглосуточный режим обработки измерений траекторий ИСЗ. Они использовались, начиная с 1957-го года при запуске первых и последующих ИСЗ, при полете Ю.А. Гагарина в 1961-м году и затем в течение последующих 10 лет. Значение этих работ трудно переоценить, потому что результаты траекторных расчетов, производимых в разных организациях, иногда не совпадали, что для управления космическими полетами было недопустимо.

Весьма значительным было влияние ИПМ и лично М.Р. Шуры-Буры на выбор архитектуры отечественных универсальных

компьютеров. В 1955-м году на начальной стадии проекта ЭВМ первого поколения М-20 в разработке участвовали три человека: С. А. Лебедев (общие характеристики и структура машины), М.Р. Шура-Бура (система команд), П.П. Головистиков (схемотехника). Основные архитектурные решения М-20, предложил М.Р. Шура-Бура. Эти архитектурные решения М-20 были сохранены в ЭВМ М-220, М222, построенных на основе полупроводниковой элементной базы. Эти машины стали ***«рабочими лошадками»*** для выполнения научных и инженерных расчетов ***во многих исследовательских, проектных и оборонных организациях страны.*** Это была одна из немногих моделей ЭВМ, при создании которой ***объединились проектанты, конструкторы и математики,*** представленные ИТМ и ВТ, конструкторским бюро, создавшим машину «Стрела».

Эта солидная основа возлагала большую ответственность на разработчиков, поскольку ее архитектуре предстояло воплотиться в нескольких крупных сериях ЭВМ (М-20, БЭСМ-4, М-220). Для машин типа М-20 – БЭСМ-4, которая также относилась к семейству машин С.А. Лебедева, было разработано, по крайней мере, три системы технологических программ в ИПМ АН СССР, в МГУ, в СО АН СССР. Эти системы ***отличались мнемоникой*** задания кодов операций, методами кодирования адресных полей машинных команд и методами настройки программ при размещении их в памяти. В это же время велись интенсивные работы по созданию систем библиотечных программ, отличавшихся друг от друга по правилам размещения их в оперативной памяти и по механизмам обращения к ним. В автокодах учитывалась необходимость размещения библиотечных программ в любом месте оперативной памяти, и были разработаны механизмы настройки подпрограмм по адресам размещения. При проектировании архитектуры машин предусматривалась аппаратная поддержка механизмов обращения к подпрограммам (процедурам) и методов передачи параметров.

В сферу научных исследований и разработок в начале 60 – х годов в Советском Союзе (почти одновременно в несколько ином виде в США) вошел и был апробирован ***новый широкий класс вычислительных систем и телекоммуникационных сетей реального времени – первый советский прототип*** современных информационных глобальных сетей и ***Интернета.*** В нем основными компонентами и источниками информации являлись траектории воздушных объектов, характеризующиеся их назначением, координатами и обобщенными параметрами движения, определяющие требования к функциям сложных

комплексов программ управления в системе противовоздушной обороны (ПВО) [12]. Телекоммуникационные сети ЭВМ обеспечили обмен и обобщение информации от радиолокационных узлов на большой территории страны для непрерывного обнаружения и сопровождения воздушных объектов. К таким системам заказчиком предъявлялись высокие требования к качеству функционирования и гарантированного решения задач.

Примером оригинальных (в то время секретных) работ в НИИ-5 (МНИИПА) являлось создание программ реального времени и телекоммуникационной сети системы ПВО страны и радиолокационного узла (РЛУ) «Межа» (главный конструктор Владимир Алексеевич Шабалин, заместитель – Анатолий Николаевич Коротоношко). Программный комплекс обработки радиолокационной информации в 1962-м – 68-м годах на ЭВМ 5Э89 был создан под руководством Владимира Васильевича Липаева (докторская диссертация – 1967-й год по специальности радиолокация). При этом был разработан в 1962-м году принципиально ***новый тип операционной системы реального времени*** на ЭВМ для автоматической синхронизации и управления динамическим решением разнородных задач о движущихся воздушных объектах при случайных потоках информации из внешней среды и случайной длительности обработки каждого сообщения. Операционная система обеспечивала функционирование комплекса программ ***телекоммуникационные сети*** для транспортировки и обработки информации на ЭВМ между несколькими соседними РЛУ о траекториях движения динамических объектов и для обобщения характеристик их траекторий.

В эти годы генерирование динамических тестов от внешних объектов ***на специализированных мобильных ЭВМ было невозможно*** вследствие ограниченности их вычислительных ресурсов. В 1965-м году для имитации тестов от движущихся объектов внешней среды ***в реальном времени*** были разработаны ***программы формирования магнитофильмов*** на универсальной ЭВМ М-20. На этой машине предварительно формировались и записывались на специализированных магнитофонах ***наборы динамических тестов о разнообразных ситуациях воздушной обстановки и движения объектов с регистрацией значений реального времени*** сообщений и их координат с точностью до секунды. ***Имитации внешней среды и динамических тестов в реальном времени,*** впоследствии стало широко применяться при разработке комплексов программ оборонных систем для испытаний и гарантирования их качества.

Возрастание ***сложности и ответственности оборонных зада*** ч, которые решаются крупными системами, а также увеличение возможного ущерба от недостаточного качества комплексов программ, значительно повысило актуальность освоения методов стандартизированного описания требований, а также оценивания характеристик качества на различных этапах жизненного цикла сложных комплексов программ. Широкое многообразие классов и видов программ, обусловленное различными функциями оборонных систем, предопределяло формальные трудности, связанные с методами и процедурами ***доказательства соответствия программного продукта*** условиям контрактов, требованиям заказчиков и потребителей. По мере расширения сферы применения и увеличения сложности выделились области, в которых ошибки или недостаточное качество программ или данных могли нанести ущерб, значительно превышающий положительный эффект от их использования.

Для создания безопасных систем и программных продуктов, прежде всего, необходимо было ***формализовать их назначение, функции и основные характеристики.*** На этой основе должны разрабатываться требования к безопасности и другим характеристикам качества, к обработанной информации для потребителей, адекватной назначению и функциям систем. Требования к функциям систем и программным продуктам, а также к безопасности их функционирования должны были соответствовать доступным ресурсам для их реализации с учетом допустимого ущерба – рисков при неполном выполнении требований. Основными источниками отказовых ситуаций были некорректные исходные требования, сбои и отказы в аппаратуре, дефекты или ошибки в программах и данных функциональных задач, проявляющиеся при их исполнении в соответствии с назначением. ***Стратегической задачей*** в жизненном цикле оборонных систем стало ***обеспечение требуемого качества программных продуктов при реальных ограничениях на использование вычислительных и иных ресурсов, выделяемых для их разработки и применения.***

1.4. Организация подготовки первых программистов в 1950-е – 60-е годы

В Московском, Ленинградском и Киевском университетах в 1950-е годы началась подготовка специалистов по вычислительной математике, в технических высших учебных заведениях появились курсы по вычислительной технике, и стали открываться кафедры вычислительных машин [1, 4, 11]. Министерство высшего образования и Высшая аттестационная комиссия ввели формальный список таких специальностей. Эти списки в системе образования и научной аттестации играли в СССР важную роль, т. к. служили средством идентификации и формального признания квалификации специалистов. В частности, каждая образовательная специальность получала право иметь самостоятельный учебный план от первого до выпускного года обучения. Учебный план в своей основной части являлся обязательным для каждого вуза и утверждался министерством. Имелось, однако, некоторое количество курсов и семинаров по выбору, которые использовались для более конкретной специализации студентов в рамках данной специальности.

В 1952-м году в нескольких университетах была открыта в дополнение к существовавшей специальности «математика» новая специальность «вычислительная математика», предназначенная для подготовки специалистов, использующих вычислительную технику. ***Первый учебный курс программирования*** в СССР был прочитан А.А.Ляпуновым в 1952-м – 53-м ***учебном году.*** Структура курса складывалась на глазах у студентов. В перерыве между первым и вторым семестрами у лектора начали складываться основные подходы к «операторному методу». Вся вторая половина курса – это была по существу совместная работа профессора и студентов по созданию и уточнению символики операторов, используемых при составлении схем программ. Курс читался и воспринимался с большим энтузиазмом, и неслучайно почти половина слушателей, математиков-вычислителей, стали после выпуска профессиональными программистами. В 1955-м году в Московском университете при кафедре вычислительной математики работал семинар по смежным вопросам кибернетики и физиологии, который с 1956-го года принял название «***семинар по кибернетике».***

В 1955-м году ***чтение курса программирования*** в МГУ продолжил М.Р. Шура-Бура [1, 4]. Первой книгой об ЭВМ,

рассчитанной на массового читателя, была книга А.И. Китова «Электронные цифровые машины», вышедшая в середине 1956-го года. Хорошим качеством книги была убедительная и увлекающая свежего читателя демонстрация новизны, вносимой ЭВМ в практику человеческой деятельности. Ее развитием стал учебник А.И. Китова и Н.А. Криницкого «Электронные цифровые машины и программирование» [5]. Это была первая книга, официально рекомендованная министерством высшего образования ***в качестве учебного пособия,*** весьма солидного объема (572 стр.), и изданная большим тиражом (25 тыс. экземпляров). ***Первым учебником,*** специально посвященным программированию, была книга киевских авторов Б.В. Гнеденко, А.С. Королюка и Е.Л. Ющенко «Элементы программирования». Они использовали для изложения условную ЭВМ и дидактику курса А.А. Ляпунова. Отдельная глава была посвящена символике адресного программирования. Первой попыткой создать солидный университетский курс программирования, базировавшийся на Алголе-60, была книга Е.А. Жоголева и Н.П.Трифонова «Курс программирования», основанная на опыте чтения лекций по программированию в МГУ.

Потребности в специалистах по программированию и в усилении подготовки по технологии системного программирования, как для общего математического обеспечения, так и для прикладных программ, привели к организации в 1969-м году ***новой специальности «прикладная математика»*** (для университетов и политехнических институтов), а также специальности «автоматизированные системы управления» (АСУ) (для отраслевых институтов). В 1975-м году подготовка по этим специальностям осуществлялась на 54 (прикладная математика) и 43 (АСУ) факультетах с общей численностью выпуска порядка 5000 человек в год.

Глава 2. История отечественной вычислительной техники в 1950-е – 70-е годы

2.1. История семейства стационарных универсальных вычислительных машин «Урал» в 1960-е – 70-е годы

В середине 60-х годов и в ***последующие*** годы, заводами страны производился серийно ***ряд оригинальных типов универсальных ЭВМ—*** БЭСМ-4; Урал-11 – 14; М-220; М-222; Минск-22; Минск-32; Раздан-2; Наири; Мир-1– 3 и другие – (см. рис. 1). Наиболее полно перечень свыше тридцати типов и десяти семейств ЭВМ, разработанных в СССР, представлен в Виртуальном компьютерном музее [10]. Некоторые ЭВМ имели экспериментальный характер или выпускались столь малыми сериями, что практически не отражались на вычислительном потенциале страны и не позволяли широко распространять и применять разрабатываемые на них программы. Поэтому далее в монографии этапы история программной инженерии отражены на ряде примеров технологических программных средств и операционных систем, оказавших наибольшее влияние на вычислительный потенциал страны, так как было нецелесообразно излагать историю программной инженерии для всей совокупности созданных вычислительных машин.

Для выделенных и рассматриваемых машин были созданы различные по функциям и качеству операционные системы и технологические компоненты программной инженерии. Квалификация их разработчиков значительно различалась, среди их продуктов можно найти оригинальные технические решения, однако большинство обеспечивало основные типовые функции автоматизации программирования, для более или менее комфортного применения соответствующих ЭВМ индивидуальными пользователями. В конце 60-х годов стало ясно, что необходимо сокращать разнотипность машин и сосредоточить их производство и разработку технологического программного обеспечения на нескольких типах наиболее перспективных универсальных ЭВМ для массового применения в научных учреждениях и промышленных предприятиях страны. Для таких ЭВМ следовало активизировать и сконцентрировать усилия специалистов по их оснащению эффективными средствами программной инженерии с целью расширения сфер применения и повышения производительности разработчиков прикладных программных продуктов.

Пензенская научная школа в области вычислительной техники, созданная Баширом Искандеровичем Рамеевым

получила широкую известность и признание благодаря его таланту и колоссальному труду, вложенному в разработку и выпуск целого ряда вычислительных машин [11]. Первый, ламповый «Урал -1», был выпущен в 1957 году. Он стал «рабочей лошадью» во многих вычислительных центрах страны. Для серийного производства машины «Урал-1» был выбран завод в Пензе. Вместе с группой молодых специалистов, работавших с ним в Москве в СКБ-245, Б.И. Рамеев в 1955 – м году переехал в этот город. Коллектив разработчиков, который составил Пензенскую школу, начал складываться в 1952 – 54 годах еще в Москве в СКБ-245. Часть сотрудников училась в МИФИ, а после окончания института были направлены в СКБ-245.

В Пензе Б.И. Рамеев становится главным инженером и заместителем директора по научной работе НИИ математических машин (потом НИИ управляющих машин) и главным конструктором вычислительных машин «Урал». Машина «Урал-1» стала родоначальницей целого семейства. Простота машины, удачная конструкция, невысокая стоимость обусловили ее широкое применение. После «Урал-1» на той же элементной базе (на электронных лампах) были созданы еще две машины: в 1959 году – «Урал-2», а в 1961 – м году – «Урал-4». По сравнению с первым «Уралом» их быстродействие увеличилось в 50 раз, оперативная память была реализована на ферритовых сердечниках и значительно увеличен объем внешней памяти.

В 1960-м году были начаты работы по созданию семейства полупроводниковых «Уралов». Основные черты нового поколения машин были сформулированы еще в 1959-м году. В соответствии с ними определили состав семейства машин, их структуру, архитектуру, интерфейсы, установили принципы унификации, утвердили технические задания на устройства, ограничения на используемые комплектующие изделия и некоторые другие документы. В процессе проектирования обсуждались с разработчиками основные решения и ход работы. В ноябре 1962-го года была закончена разработка унифицированного комплекса компонентов «Урал-10», рассчитанного на автоматизированное производство. Хотя компоненты разрабатывались для использования в серии ЭВМ «Урал-11» – «Урал-16», они нашли широкое применение и в других средствах вычислительной техники и автоматике. Для этих целей было выпущено несколько миллионов штук компонентов.

В семейство полупроводниковых «Уралов» входили три модели: «Урал-11», «Урал-14» и «Урал-16». Первые две модели стали выпускаться серийно с 1964 года, а последняя – с 1969 года. Выпуск моделей этого семейства ознаменовал новую веху в

творческом наследии главного конструктора Б.И. Рамеева. Это первое в нашей стране семейство машин с унифицированной системой организации связи с периферийными устройствами (унифицированный интерфейс), унифицированной оперативной и внешней памятью. В моделях этого семейства нашли свое воплощение многие идеи, которые затем широко использовались в машинах третьего поколения (развитая система прерываний, эффективная система защиты памяти, развитое программное обеспечение).

Это семейство являлось выдающимся примером создания ***массовых, программно совместимых универсальных ЭВМ разной мощности в 70-е голы***, на единой конструктивной, технологической и схемной базе. ***Основные особенности поколения машин,*** воплощенные Б.И. Рамеевым в серии «Урал», сводились к следующему:

• машины представляли собой конструктивно, схемно– и программно совместимый ряд ЭВМ различной производительности, с гибкой блочной структурой;

• с широкой номенклатурой устройств, со стандартизованным способом подключения, позволяющим подобрать комплект машины, наиболее подходящий для данного конкретного применения, и поддержать в процессе эксплуатации параметры машины на уровне изменяющихся потребностей заказчика и новых разработок устройств;

• конструктивные и схемные возможности позволяли комплектовать системы обработки информации, состоящие из нескольких одинаковых или разных машин, обеспечивая плавное изменение количественных характеристик и существенно расширяя ряд в сторону увеличения производительности, круга решаемых задач и областей применения;

• наличие датчика времени, аппаратуры сопряжения с каналами связи и пультов операторов для связи с машиной давали возможность строить различные системы обработки данных коллективного пользования, работающие в режиме разделения времени;

• возможности резервирования отдельных устройств и машин обеспечивали создание систем повышенной надежности для обработки информации в заданное время.

В ***семействе ЭВМ были предусмотрены:***

система схемной защиты информации, независимость программ от места в памяти, система относительных адресов, развитая система прерываний и приостановок и соответствующая система команд, позволяющая организовать сложную систему одновременно работающих устройств и одновременное решение

многих задач;

• – возможность резервирования отдельных устройств машин, позволяющая создавать системы повышенной надежности: системы схемной защиты данных, независимость программ от их места в памяти, система относительных адресов, развитая система прерываний и соответствующая система команд;

• возможность работать в режимах: с плавающей и фиксированной запятой, в двоичной и десятичной системах счисления, выполнение операций со словами фиксированной и переменной длины, что позволяло эффективно решать, как планово-экономические, информационные, так и научно-технические задачи;

• система аппаратного контроля устройств хранения, адресации, передачи, ввода и обработки информации;

• большая емкость оперативной памяти с непосредственной выборкой слов переменной длины, эффективные аппаратные средства контроля и защиты программ друг от друга, ступенчатая адресация, развитая система прерываний и приостановок;

• возможность подключения памяти большой емкости с произвольной выборкой на магнитных барабанах и дисках, наличие датчика времени, аппаратуры сопряжения с каналами связи и пультов операторов для связи с машиной, что давало возможность строить различные системы обработки информации коллективного пользования, работающие в режиме разделения времени.

Основные черты этого поколения машин были изложены еще в 1963-м году в проекте на семейство ЭВМ. Он появился на полтора года раньше публикаций об американском семействе машин IBM-360. Таким образом, идея создания семейства программно и конструктивно совместимых ЭВМ была ***опубликована Б.И. Рамеевым независимо от американских ученых и реализована практически одновременно.*** В отличие от первых моделей семейства IBM-360, семейство «Урал» обеспечивало возможность создания систем обработки информации, состоящих из нескольких одинаковых или разных машин, было рассчитано на работу в сетях и, наконец, было открытым для дальнейшего наращивания технических средств для конкретных систем. Семейство этих ЭВМ производилось серийно с 1964-го года и более десятка лет широко применялось на промышленных предприятиях в стране.

2.2. История стационарных универсальных, высокопроизводительных ЭВМ в 1960-е – 70-е годы

Наибольшее влияние на программирование в 70-е годы оказало появление машины БЭСМ-6 [2]. Ее автором был академик Сергей Алексеевич Лебедев – глава выдающейся отечественной научной школы в области вычислительной техники и программирования. В архитектуре этой машины было сделано много для аппаратной поддержки операционных систем: аппаратная поддержка виртуальной памяти; защита памяти; развитая структура двухуровневой системы прерываний; защищенный супервизорный режим и т. п. Все эти характеристики являлись неотъемлемым признаком современных процессоров, но во времена создания БЭСМ-6 это было необычным и новым. БЭСМ-6, разработанная в ИТМ и ВТ совместно с Московским заводом счетно-аналитических машин (САМ), начала выпускаться с 1968 года, а в 70-х годах была среди универсальных ЭВМ самой высокопроизводительной в мире.

Основная цель [2, 7], которую преследовали авторы проекта БЭСМ-6 – создать быстродействующую серийную машину, сравнительно дешевую, удовлетворяющую наиболее важным современным требованиям с точки зрения автоматизации программирования и развития операционных систем, оснащенную имевшимися в то время в отечественном серийном производстве внешними запоминающими устройствами и устройствами ввода-вывода. Машина БЭСМ-6 предназначалась ***для решения крупных научно-технических задач,*** что, естественно, отразилось как на ее архитектуре, так и на выборе системы элементов и конструкции. Она не являлась копией какой-либо отечественной или зарубежной установки ни по системе команд, ни по внутренней структурной организации. При ее создании и проектировании был изучен и проанализирован опыт создания ЭВМ высокой производительности, накопленный к тому времени. В БЭСМ-6 были реализованы ***новые архитектурные и схемотехнические решения,*** многие из которых отразились в появившихся потом машинах третьего поколения.

Машины БЭСМ-6 ***составили стратегическую основу*** вычислительных средств большинства крупных вычислительных центров и оборонных предприятий страны. Сфера использования машины превзошла прогнозы ее разработчиков. Первоначально

предполагалось, что небольшая серия БЭСМ-6 будет использована для решения крупных научных задач в нескольких научных институтах Советского Союза и ядерных центрах. Реально эта машина нашла значительно более широкое применение. На основе БЭСМ-6 были созданы центры коллективного пользования, центры управления в реальном масштабе времени, координационно-вычислительные центры, системы телеобработки и т. д. Машина БЭСМ-6 широко использовались в системах автоматизации проектирования, для моделирования сложнейших физических процессов и процессов управления, как инструментальная машина для разработки ***крупных программных продуктов оборонных систем*** и различных новых ЭВМ.

Важной особенностью машины явились аппаратные и программные средства для ***обеспечения мультипрограммного режима.*** К ним относятся виртуальная адресация памяти со страничной организацией, система прерывания и соответствующие программы операционной системы, наличие нескольких режимов выполнения команд в процессоре. Высокая скорость преобразования виртуальных адресов в физические обеспечивалась размещением таблицы их соответствия в регистровой памяти. Имелись аппаратные механизмы защиты памяти для команд и операндов. Все это обеспечивало возможность динамического распределения памяти в процессе вычислений средствами операционной системы.

По уровню производительности и степени согласования аппаратных средств с архитектурой, а также архитектуры – с алгоритмами научно-технических задач, БЭСМ-6 может быть отнесена к ***классу суперЭВМ.*** БЭСМ-6 за счет многочисленных нововведений архитектурного и структурного плана при основной тактовой частоте 10 МГц выполняла в среднем один миллион операций в секунду над 48-разрядными операндами. В начале 60-х годов отечественной промышленностью были созданы высокочастотные транзисторы и диоды, на основе которых была разработана элементная база машины (в машине было использовано около 60 тыс. транзисторов и 180 тыс. диодов).

Назначение машины, ее архитектурные и структурные особенности, отвечающие современным идеям, потребовали создания ***соответствующей операционной системы и системы программирования,*** удовлетворяющих требованиям пользователей. БЭСМ-6 стала первой отечественной ЭВМ, которая была принята государственной комиссией и поставлялась как система аппаратных средств ***совместно с ее системным***

программным обеспечением (см. главу 3). Работы по исследованию и разработке операционных систем, стратегий распределения ресурсов и планирования вычислений в нашей стране начались широким фронтом с появлением БЭСМ-6 [7, 11].

В 1968-м году на Московском заводе счетно-аналитических машин (САМ) началось производство ЭВМ БЭСМ-6. Полностью новый компьютер на основе транзисторов и интегральных схем был разработан под руководством С.А. Лебедева, В.А. Мельникова и Л.Н. Королева. При его разработке была поставлена серьезная задача – достичь производительности порядка 1000000 операций в секунду (один мегафлоп). БЭСМ-6 сильно опередила свое время, став началом второго поколения ЭВМ. Она вобрала в себя много оригинальных идей. Систем подобного класса в мире не было. Одно из основных отличий и главных новшеств – ***«лебедевская водопроводная структура»*** процессора, позволяющая совмещать обработку различных команд на разных стадиях их выполнения.

Уже позже западные коллеги придумали для этого метода термин ***«конвейер»*** (все процессоры на сегодняшний день используют конвейерную архитектуру). Отныне к разным блокам памяти можно было обращаться одновременно, появился прообраз кэш-памяти – сверхбыстрое устройство хранения часто используемых данных и команд. Все эти улучшения обеспечили качественный скачок производительности. Коллеги и современники ***называли С.А Лебедева настоящим гением—*** при всей сложности собранной системы, он сумел отсечь все ненужное, оставив самые необходимые блоки. В период с 1968-го по 1987-й год было выпущено порядка 400 машин БЭСМ-6, которые использовали в самых разных, ***преимущественно оборонных отраслях.*** Важной особенностью БЭСМ-6 считается программное обеспечение – впервые с момента появления отрасли, ЭВМ начали поставлять с необходимым софтом прямо с завода. Для БЭСМ-6 была разработана полноценная операционная система, над ней трудились лучшие советские умы из Института прикладной математики АН СССР, Вычислительного центра Академии наук и Московского государственного университета (см. главу 3).

С.А. Лебедев одним из первых понял значение системного программирования, значение совместной работы программистов-математиков и инженеров при создании вычислительных систем, включающих как неотъемлемую часть технологическое программное обеспечение, состав и качество которого определяет удобство использования и эффективность работы систем в целом. По инициативе С.А. Лебедева в ИТМ и ВТ в 60-е годы была создана лаборатория математического обеспечения, выполнявшая

разработку системного программного обеспечения для всех вычислительных систем: ЭВМ БЭСМ-6, многомашинного информационно-вычислительного комплекса АС-6, ЭВМ серии «Эльбрус», ЭВМ специального назначения.

Математики-программисты принимали ***полноправное участие в разработке архитектур создаваемых машин,*** математическом моделировании их структурной организации, создании системы автоматизации проектирования ЭВМ. Все схемы БЭСМ-6 по инициативе С.А. Лебедева были записаны формулами булевой алгебры, что открыло широкие возможности для автоматизации проектирования и подготовки монтажной и производственной документации. Она выдавалась на завод в виде таблиц, полученных на «инструментальной» ЭВМ БЭСМ-2. В разработке БЭСМ-6 были впервые применены методы проектирования и описания, которые в дальнейшем стали широко использоваться при создании новейших суперЭВМ. В годы становления вычислительной техники ***далеко не у всех было понимание важности системного, технологического программного обеспечения и системного (инженерного) программирования.*** Программирование систем жесткого реального времени (например, в системах ПВО) являлось одной из самых сложных задач программирования, с наиболее высокой ценой каждой допущенной ошибки, которые, тем не менее, проявлялись даже при натурных испытаниях. Надежность в значительной степени обеспечивается большим запасом мощности основных элементных блоков (диоды и транзисторы были нагружены на 25–40 % от допустимого номинала). Время наработки на отказ достигало нескольких сотен часов.

Влияние машины БЭСМ-6 на развитие отечественной вычислительной техники определялось не только длительностью эксплуатации, сколько тем, что заложенные при создании машины идеи оказались весьма плодотворными. Несколько поколений инженеров и программистов, работавших на БЭСМ-6, были воспитаны на этих идеях. Разработка БЭСМ-6, ***составившей целую эпоху в отечественном вычислительном машиностроении,*** явилась примером творческого подхода к созданию ЭВМ, учитывающего все возможности, предоставляемые технической базой, математическим моделированием структурных решений, а также возможности производства для достижения наилучших характеристик машины.

Система АС-6 (главные конструкторы – В.А. Мельников, А.А. Соколов) была ***предназначена для решения больших научных и экономических задач, задач обработки информации и управления в реальном времени*** [2, 11]. Машина разработана

коллективом ИТМ и ВТ АН СССР совместно с заводом САМ. Разработка АС-6 была завершена в 1975 году, а в 1977 году Московский завод САМ начал изготовление системы малой серией.

2.3. История стационарных, специализированных ЭВМ реального времени в 1970-е – 80-е годы

На 1970-е – 80-е годы пришлось активное развитие отечественных, специализированных ЭВМ и сложных комплексов программ для стационарных систем противоракетной обороны. При этом основное внимание было сосредоточено на разработке аппаратуры ЭВМ и на достижении высоких характеристик по их производительности в реальном времени. При доступной элементной базе это достигалось в значительной степени путем специализации архитектуры и структуры команд ЭВМ, в соответствии с конкретными функциональными задачами и алгоритмами работы оборонных систем. Высокие требования руководства страны к срокам и темпам разработки систем, приводили к сосредоточению всех усилий специалистов и промышленности на создании аппаратуры ЭВМ. До завершения ее монтажа и испытаний, программирование и отладка комплексов программ зачастую оказывалась невозможной, в частности, вследствие уникальности систем команд этих машин. Первичная разработка и отладка программ обычно начиналась ***на «сырых» машинах в объектном коде,*** практически без применения технологического инструментария ***«на одном энтузиазме».*** Впоследствии машины оснащались минимумом технологических средств на уровне автокодов, которые применялись при развитии и совершенствовании «унаследованных» комплексов программ. В результате, существовавшие в стране на других оборонных предприятиях, ***методы и инструментальные средства программной инженерии в рассматриваемой сфере в это время практически не использовались***.

В 1964 году под руководством Сергея Алексеевича Лебедева была разработана и прошла межведомственные испытания ЭВМ 5Э92б, (первая группа специализированных машин) предназначенная для использования в ***системе контроля космического пространства и обработки телеметрии спутников*** [2, 9, 11]. ЭВМ 5Э92б – модификация М-50, применялась в вычислительных и управляющих информационных комплексах управления космическими объектами, центрах контроля космического пространства. Межведомственные испытания комплекса из восьми машин прошли в 1967 году. ***Программное обеспечение включало*** развитую систему тестовых и диагностических программ, существенно использующую

аппаратный контроль и позволяющую определить неисправный блок.

Первая очередь системы включала одномашинный вычислительный комплекс 5Э92б, системы передачи данных и одного рабочего места оператора на командном пункте. В 1969 году были проведены государственные испытания, и работы по первой очереди завершились. На этом этапе вычислительный комплекс позволял ежесуточно обрабатывать около 4000 радиолокационных измерений и около 200 оптических наблюдений и иметь главный каталог емкостью до 500 формуляров по космическим объектам. Запаздывание в обработке информации было сокращено с нескольких суток до нескольких часов.

Машина 5Э92б была модернизирована в части введения арифметики с плавающей запятой и мультипрограммного режима, и получила название ЭВМ 5Э51. Ее серийный выпуск начался ***в*** 1967 году. Благодаря автономной работе основных устройств и, в первую очередь, процессора ввода-вывода на базе общего ОЗУ, эти машины успешно использовались при создании многомашинных комплексов с единой внешней памятью. ЭВМ была надежной, достаточно производительной и удобной в эксплуатации. В общей сложности Загорский электромеханический завод выпустил большую ***серию из почти трехсот ЭВМ*** 5Э92б и 5Э51. Но, несмотря на это, машин не хватало, и, как правило, все ЭВМ забирало министерство обороны. Функциональные программы системы создавались в основном в машинных кодах, а позднее на автокоде.

Вторая очередь системы предусматривала замену 5Э92б на модернизированный четырехмашинный комплекс на базе ЭВМ 5Э51. Командный пункт оборудовался коллективными средствами отображения космической обстановки и рабочими местами операторов. В 1972 году был испытан трехмашинный вычислительный комплекс 5Э51, а в 1973 году были успешно проведены испытания четырехмашинного комплекса на базе этой же ЭВМ и новой программно-алгоритмической системы, в которой обработка координатной информации была практически полностью автоматизирована. Для контроля космического пространства с командным пунктом взаимодействовали несколько измерительных пунктов (ИП) на территории СССР и на кораблях в море. На ИП предварительно, на ЭВМ М-220 обрабатывалась информация о координатах, параметрах и состоянии космических объектов, которая селектировалась и сжималась для последующей передачи на командный пункт. На этом завершились работы по созданию второй очереди. Общая

производительность вычислительного комплекса командного пункта составила около ***двух миллионов операций в секунду.*** Обладая высокой надежностью, он проработал до начала 90-х годов.

В 1957-м году началась разработка одной из первых в Советском Союзе транзисторных машин – для обработки данных радиолокационных станций (РЛС) под руководством Михаила Александровича Карцева (вторая группа специализированных машин). В ноябре 1962-го года вышло ***постановление правительства*** о запуске М-4 в серийное производство [11]. Это была первая опытная машина, сделанная на транзисторах. Еще в 1966-м году М.А. Карцев выдвинул идею создания многомашинного вычислительного комплекса, построенного из вычислительных машин, специально разработанных для совместной работы в таком комплексе. Проведенные исследования показали, что производительность комплекса может достигнуть миллиарда операций в секунду. На то время ни одна из машин в мире не имела такой производительности! Это воодушевляло М.А. Карцева, увлекало коллектив разработчиков. Уже в 1967-м году был разработан эскизный проект комплекса (ВК М-9). При защите в министерстве он получил положительную оценку. Вскоре был организован Научно-исследовательский институт вычислительных комплексов (НИИ ВК), а самого М.А. Карцева назначили директором.

В марте 1963-го года распоряжением Военно-промышленной комиссии НИИ ВК была поручена разработка ЭВМ М4-2М. Задача системы – обеспечивать военно-политическое руководство страны достоверной информацией о возможной угрозе ракетного нападения и обстановке в космосе, т. е. она имеет чисто оборонительный характер. Для обеспечения возможности работы на трех уровнях были созданы три модификации М4-2М – 5Э71 для радиолокационных станций, 5Э72 – для командных пунктов (КП) радиолокационных узлов, 5Э73 – для будущего КП комплекса ПРО. В октябре 1964 года начались приемо-сдаточные испытания, в ноябре машина была принята заказчиком и отправлена на головной объект. К концу 1964-го года еще шесть машин 5Э71 были отгружены на объекты заказчика. В 1965-м – 66-м годах были проведены стыковки машин с РЛС и отработка на них программного обеспечения, ***которое создавалось следующим образом*** [9, 11].

Работы шли в три смены, особенно у разработчиков программ, которые отлаживались на штатных ЭВМ в составе РЛС и командного пункта ПРО. Машинное время расписывалось до минуты. Программирование рабочих алгоритмов было очень

трудоемким процессом. ***Средств автоматизации программирования в то время практически не было***, да и применить их было нельзя. Это было неизбежно из-за крайне ограниченных ресурсов памяти и производительности ЭВМ. Чтобы «втиснуть» функциональные программы системы в отведенные ей память и время работы, приходилось перепрограммировать по несколько раз. Современные программисты вряд ли могут представить, как можно было сжать до четырех тысяч машинных команд всю рабочую программу радиолокационного узла ПРО. Он с двумя секторными РЛС должен был обеспечивать одновременное обнаружение траекторий целей, уточнение траекторий для шести сопровождаемых целей, выдачу в нужной форме информации на средства отображения и на командный пункт.

В 1963-м году в тематическом отделе была создана лаборатория по разработке программного обеспечения для будущих радиолокационных станций и командных пунктов узлов ПРО. В лаборатории было определено ***два направления по созданию программно-алгоритмического обеспечения.*** Разработка алгоритмов для командных пунктов узлов была поручена одной группе специалистов, а разработка алгоритмов и программ для радиолокационных узлов другой. Главная трудность на первом этапе заключалась в том, чтобы представить себе поставленную задачу, понять, что нужно делать и как ее решать. Ни в зарубежной, ни в отечественной закрытой литературе найти прототипы не удалось (последствия барьеров секретности). Прежде всего, были определены алгоритмы, связанные с траекторной обработкой информации о космических объектах. С большим трудом удавался выбор методов первичной обработки радиолокационной информации от момента выхода ее с аппаратуры РЛС до момента формирования математических опорных точек траекторий целей. (Однако, ***в то же время*** подобные задачи успешно решались в НИИ-5 при создании радиолокационных узлов «Межа» и командных пунктов системы ПВО страны, см. главу 3. Эти проекты, к сожалению, ***вследствие глубокой секретности, развивались параллельно и совершенно независимо,*** без обмена информацией.

В ***конце*** 1965-го года на узле были смонтированы две из трех предусмотренных по штату ЭВМ 5Э71, функционировало несколько линеек приемно-индикационной аппаратуры станции, настраивались передатчики. Можно было начинать отладку боевой программы на реальных вычислительных средствах и проводить пробную стыковку с аппаратурой [9]. ЭВМ 5Э71 имела высокую надежность, была простой и удобной в эксплуатации, но

для программистов была пока еще фактически грудой железа. ***Никаких технологических программных средств к ней не прилагалось.*** Не было даже операционной системы реального времени и программной среды, в которой она работает. Все это предстояло создавать на месте. Год непрерывной круглосуточной работы программистов и алгоритмистов принес свои плоды. К концу 1966 года боевая программа уже вполне достойно функционировала в составе системы. Формальным и успешным подтверждением этому были завершившиеся конструкторские испытания станции. Существенных претензий к боевой программе ни со стороны главного конструктора, ни со стороны заказчиков объекта предъявлено не было. ***(РЛУ «Межа»*** на ЭВМ 5Э89 в системе ПВО ***успешно завершил государственные испытания в*** 1967 году).

В 1969-м году главный конструктор М.А. Карцев ***начал разработку*** ЭВМ М-10, которая в 1973-м году начала эксплуатироваться [3, 9]. Создание ЭВМ М-10 долго держалось в глубоком секрете, потому что машина разрабатывалась для Системы предупреждения о ракетном нападении (СПРН), а также для общего наблюдения за космическим пространством. На околоземных орбитах тогда находится около 17 тысяч объектов различного происхождения, включая действующие и отслужившие свой срок спутники, куски ракетоносителей и пр. Первый эшелон СПРН – космический: по факелам запускаемых ракет, на спутниках засекают их старт. Костяк системы – ее второй, наземный эшелон, включающий мощные радиолокационные станции, расположенные по окраинам страны (до развала СССР их было девять – под Ригой, Мурманском, Балхашем, Иркутском, и т. д.), а также сопряженные с ними вычислительные комплексы на базе ЭВМ М-10, которые имели следующие ***основные характеристики*** [3]:

• среднее быстродействие – 5 млн. операций в секунду, быстродействие на малом формате <16 разрядов – около 10 млн. операций в секунду;

• общий объем внутренней памяти – 5 млн. байт;

• первый уровень – оперативная память – 0,5 млн. байт; постоянная память 0,5 – млн. байт.

• второй уровень памяти – 4 млн. байт;

• показатели надежности: коэффициент готовности – не менее 0,975, время (среднее) безотказной работы – не менее 90 часов;

• обеспечивалась одновременная работа 8 пользователей на восьми математических пультах.

К началу 1980-х годов ЭВМ М-10 обладала: наивысшими производительностью (по некоторым оценкам – 20–30 млн.

операций в сек.), емкостью внутренней памяти и пропускной способностью мультиплексного канала, достигнутыми в СССР. Впервые в мире в ней был реализован ряд новых прогрессивных решений, в том числе: предусмотрена возможность синхронного комплексирования до семи ЭВМ при прямом (минуя мультиплексный канал) обмене информацией между программами отдельных машин. При динамическом разделении оборудования; реализована автоматическая перестройка поля процессоров; в состав ЭВМ введен второй уровень внутренней памяти емкостью более 4 млн. байт с произвольным доступом; обеспечен внешний обмен с обоими уровнями внутренней памяти.

В 1978-м году главный конструктор М-10 – М.А. Карцев, предложил приступить к работам по созданию ***новой, многопроцессорной векторной вычислительной машины*** М-13, используя опыт, полученный при разработке, изготовлении и эксплуатации машин М-10 и М-10М, а также новейшие достижения в технологии и в электронной технике [3, 11]. В 1979-м году коллектив НИИ ВК начал разработку конструкторской документации. Были определены и заводы-изготовители, на которых предполагалось вести производство машины М-13. В течение 1980-го – 81-го годов конструкторская документация комплектно, по устройствам была передана на заводы.

ЭВМ М-13 ***стала машиной четвертого поколения,*** в качестве элементной базы в ней были использованы большие интегральные схемы. В архитектуре этой многопроцессорной векторной ЭВМ, предназначенной, в первую очередь, для обработки в реальном масштабе времени больших потоков информации, были предусмотрены четыре основных части: центральная процессорная часть, аппаратные средства поддержки операционной системы, абонентское сопряжение, специализированная процессорная часть. Специализированная процессорная часть машины была предназначена для обработки больших массивов относительно малоразрядной информации (быстрое преобразование Фурье, вычисление корреляционных функций, сравнение с порогом, проверка гипотез и др.) и имела в качестве базовых операции произведение двух комплексных чисел (двухточечное преобразование Фурье). Специальный (комплексный) арифметический процессор выполняет эту базовую операцию за один машинный такт. Эквивалентное быстродействие линии комплексных процессоров на порядок превышало быстродействие линии арифметических процессоров на сопоставимых форматах данных. Эквивалентное

быстродействие специализированной процессорной части машины М-13 в максимальной комплектации при решении указанных выше задач могло достигать $2{,}410^9$ операций в секунду.

Абонентское обеспечение машины М-13 содержало операционную систему, систему программирования и отладки, файловую систему, систему документирования, библиотеку типовых программ и обеспечивало [3, 11]:

- реальный масштаб времени (РМВ), режим разделения времени (РВ), пакетную обработку;
- 4 задания РМВ, 16 заданий РВ;
- многосеансовое выполнение до 256 заданий;
- устранение последствий сбоев и резервирование.

Система автоматизации программирования и отладки включала:

- ассемблеры, Т-язык;
- алгоритмический язык высокого уровня, ориентированный на векторные вычисления;
- интерактивный режим отладки заданий РВ и РМВ в понятиях используемого языка;
- файловую систему;
- систему документирования;
- библиотеку типовых программ;
- систему технического обслуживания.

Машина М-13 имела модульное построение и допускала переменную комплектацию, способную оптимально обеспечить пользователю необходимые технические характеристики. Центральная процессорная часть имела три конфигурации и могла иметь производительность в зависимости от исполнения $12^* 10^6$; 24^*10^6 и $48^* 10^6$ операций в секунду. При этом также соответственно изменялся и объем внутренней памяти, пропускная способность центрального коммутатора и пропускная способность мультиплексного канала. Объем внутренней памяти мог составлять 8,5; 17,0 или 34,0 Мбайт, пропускная способность центрального коммутатора – 800; 1600 или 3200 Мбайт/сек., пропускная способность мультиплексного канала – 40; 70 или 100 Мбайт/сек. Эквивалентное быстродействие специализированной процессорной части машины М-13 в максимальной комплектации, при решении указанных выше задач может достигать $2{,}4^*10^9$ операций в секунду.

В многопроцессорной системе 4-го поколения М-13 впервые реализована аппаратура пооперационных циклов (обеспечивающая независимость программы от числа процессоров в системе), аппаратура сегментностраничной организации памяти

(перекрывающая возможности файловой системы), программноуправляемый периферийный процессор для операций типа преобразования Фурье, Уолша, Адамара, Френеля, вычисления корреляционных функций, пространственной фильтрации и т. п. Среднее быстродействие центральной части – до 50 млн. операций в секунду (или до 200 млн. коротких операций в секунду), внутренняя память – до 34 Мбайт, скорость внешнего обмена – до 100 Мбайт в секунду, эквивалентное быстродействие периферийного процессора на своем классе задач – до 2 миллиардов операций в секунду.

М.А. Карцев – автор фундаментальных теоретических работ по вычислительной технике (5 монографий, 16 изобретений). Книги «Арифметические устройства электронных цифровых машин» (русское издание – 1958 г., позднее переиздавалась за рубежом), «Арифметика цифровых машин» (1969 г.) заложили основы теории арифметических устройств; их выводы вошли в учебники «Архитектура цифровых вычислительных машин» и «Вычислительные системы и синхронная арифметика», где практически впервые сделана попытка поставить на научную основу проектирование общей структуры ЭВМ и аппаратуры для выполнения параллельных вычислений.

Специализированные ЭВМ реального времени (третья группа ЭВМ) МВК Эльбрус-1 (1979-й год) и МВК Эльбрус-2 (1984-й год) (С.А. Лебедев,

В.С. Бурцев), относились по существу к следующему этапу (1980-е – 90-е годы) развития специализированной отечественной вычислительной техники [2, 11]. Однако их целесообразно кратко представить в данном разделе, вследствие основной области применения. Эти МВК двойного применения (гражданского и военного), предназначались для использования в высокопроизводительных информационно-вычислительных и управляющих системах, в том числе, в системах непрерывного действия, работающих в реальном масштабе времени, а также в научных и промышленных вычислительных центрах коллективного пользования в пакетном режиме и в режиме реального времени.

Программное обеспечение являлось общим для МВК Эльбрус-1 ***и*** Эльбрус-2. Его отличительная особенность состояла в использовании языка высокого уровня ЭЛЬ-76, являющегося автокодом системы для написания системных программ, в частности, операционной системы, трансляторов и целого ряда управляющих программ, работающих в реальном масштабе времени. Это позволило значительно сократить время создания программ.

Создаваемые на базе МВК Эльбрус-2 вычислительные комплексы имели высокие показатели надежности и достоверности выдаваемой информации за счет модульного принципа построения и наличия системы реконфигурации, которая при возникновении сигнала неисправности от системы аппаратного контроля модуля, автоматически исключала его из состава комплекса и восстанавливала прерванный вычислительный процесс. Большое значение в достижении высоких показателей надежности имела система тестовых и диагностических программ. Их отличительной особенностью являлась способность обрабатывать динамические ситуации по сбоям и отказам, зафиксированные как на тестовых программах, так и на программах пользователя. Набранная статистическая информация по сбоям и отказам модуля использовалась для принятия решения о необходимости профилактики или ремонта устройства.

Опыт, полученный при создании системы ПРО, показал, что сроки разработок в значительной степени были связаны с ***временем отработки алгоритмов и программ.*** С целью совершенствования подготовки системных программистов ЦНПО «Вымпел» совместно с Сибирским отделением АН СССР в 1980 году решили создать в Новосибирском Академгородке ***конструкторское бюро системного программирования*** (КБ СП) в составе Вычислительного центра СО АН СССР. Конструкторское бюро должно было также оказать помощь ИТМ и ВТ в программировании и отладке для «Эльбруса».

Вскоре выяснилось, что сибирские академики решили использовать это КБ в своих научных целях, а практические задачи по программированию в интересах ПРО и СПРН их не интересуют. Возник конфликт [9, 11]. Для его разрешения министр радиопромышленности СССР Валерий Дмитриевич Калмыков, по договоренности с академиком Михаилом Александровичем Лаврентьевым, направил в Новосибирск двух своих заместителей. В Новосибирске уже на следующий день выявились принципиальные разногласия сторон. Тогда было предложено разделить КБ на две части, одну из которых перевести на одно из предприятий ЦНПО «Вымпел», а другую оставить под эгидой Вычислительного центра Сибирского отделения АН совместно с ИТМ и ВТ осваивать ЭВМ «Эльбрус». Это компромиссное предложение было принято. Часть коллектива КБ СП в количестве 100–120 человек перевели в Гомель. На базе этого коллектива, усиленного молодыми специалистами – выпускниками вузов, было создано первое в стране высококвалифицированное ***Конструкторское бюро системного***

программирования. Позже «Эльбрус-2» прошел испытания и поступил на вооружение. По два десятипроцессорных комплекса были установлены на объектах боевой системы – командно-вычислительном пункте ПРО.

2.4. История единого семейства (ЕС) ЭВМ в 1970-е – 80-е годы

В СССР в 1970 – 80-е годы разработкой разнообразных ЭВМ занимались ***множество различных институтов и промышленных предприятий.*** К этому периоду в СССР наблюдался бурный рост выпуска ЭВМ второго поколения [10]. Нарастала необходимость серьезной ***стандартизации*** семейств вычислительной техники, программного обеспечения, кодов, протоколов, интерфейсов. Многим руководителям промышленности и специалистам стало ясно, что необходимо ***объединение и концентрация усилий в стране*** для создания и развития современной вычислительной техники и программного обеспечения для различных отраслей народного хозяйства и систем вооружения. Для этого следовало разработать концепцию проектов, позволяющих сократить хаос и широкий спектр дублирующихся разработок разнообразных ЭВМ и компонентов программного обеспечения.

Естественно, встал вопрос ***о координации развития*** вычислительной техники и о создании унифицированных семейств ЭВМ третьего поколения, разной мощности и назначения. Для этого должна была разработана ***государственная стратегия,*** развития вычислительной техники и программного обеспечения на ближайшие десятилетия [23, 25]. Несмотря на наличие некоторых достижений в области программирования, программного обеспечения ЭВМ, выпускаемых в стране в шестидесятых годах, было катастрофически недостаточно, и серьезно сдерживало их применение в ряде отраслей народного хозяйства. Эти факты хорошо понимали многие отечественные специалисты, но на их согласование и принятие решений ушли годы дискуссий.

В 1968-м году началось в СССР практическое создание ***семейства совместимых ЭВМ общего назначения третьего поколения,*** хотя подготовка, обсуждение основных концепций, организация разработки проекта велись нарастающими темпами с 196бго года. Ниже изложены особенности развития ***трех*** семейств ***ЭВМ и их программного обеспечения,*** оказавших наибольшее влияние на вычислительный потенциал страны:

- универсальных, стационарных ЭВМ общего назначения (ЕС ЭВМ);
- мобильных, бортовых ЭВМ, совместимых с ЕС ЭВМ (БЭВМ);
- системы стационарных, малых управляющих ЭВМ (СМ ЭВМ).

В 1966-м году в народнохозяйственном плане появилось задание Минрадиопрому (МРП) СССР разработать ***аванпроект Елиной Системы ЭВМ – ОКР «Ряд».*** Задание было сформулировано начальником Главного управления по вычислительной технике МРП – Михаилом Кирилловичем Сулимом. Оно предписывало в течение 1966-го и 1967-го годов представить аванпроект «Комплекса типовых, высоконадежных информационных вычислительных машин с диапазоном по производительности от 10 тыс. до 1 млн. операций в секунду, построенных на единой структурной и микроэлектронной технологической базе, и совместимых системах программирования для вычислительных центров и автоматизированных систем обработки информации». Наибольшую активность в обсуждении проблем ряда совместимых ЭВМ проявляли Институт прикладной математики (ИПМ) АН СССР, Конструкторское бюро промышленной автоматики (КБПА), НИИсчетмаш и СКБ Минского завода им. Г.К. Орджоникидзе.

В ***январе*** 1967 года академик А.А. Дородницын, возглавлявший комиссию АН СССР и ГКНТ по вычислительной технике, выступил с докладом на коллегии ГКНТ «О состоянии математического обеспечения ЭВМ и мерах по его коренному улучшению». В нем было объективно отражено состояние с математическим (программным) обеспечением в СССР и предложены меры по его развитию. Докладчик оценил отставание от США в этой области обеспечения минимум в девять лет. Он назвал число программистов в СССР – 1500 человек, из которых 660 трудились в 18 союзных и республиканских ВЦ, в институтах – разработчиках ЭВМ, в институтах АН СССР и университетах, т. е. в основных научных организациях, занятых программированием и созданием средств его автоматизации. Численность программистов только в двух организациях – Институте кибернетики АН УССР и СКБ Минского завода им. Г.К. Орджоникидзе составляла более 100 человек, а в остальных 16 организациях не превышала 40 человек в каждой. (Однако ***на предприятиях оборонной промышленности*** в это время трудилось уже ***около 20 тыс. человек —*** разработчиков программ – см. главу 3). В то же время численность программистов в США оценивалась в 50 тыс. человек. Докладчик утверждал, что в СССР не было ни одной организации, способной в разумные сроки представить современные операционные системы (ОС) для новой серии ЭВМ, для этого не было ни людских, ни технических, ни финансовых ресурсов (но была создана мощная ОС для БЭСМ-6). Прикладные программы каждый пользователь создавал для себя

сам, поставщики ЭВМ ими не занимались, и это вызывало главную озабоченность у комиссии по вычислительной технике [10, 24].

Дискуссия в основном сводилась к тому, что желательно использовать западную систему-прототип, и возможна ли реализация архитектуры и использования программного обеспечения IBM-360 в условиях жесткого эмбарго США, ибо если это без документации и образцов невозможно, то не стоит тратить силы на точное воспроизведение и ее нужно ***«улучшать».*** Конец этой дискуссии положило решение комиссии по ВТ АН СССР и ГКНТ ***в январе*** 1967-го года, которым было предложено принять для «Ряда» архитектуру IBM-360 «с ***целью возможного использования того задела программ, который можно полагать имеющимся для системы*** IBM-360». В этих условиях обеспечение полной ***совместимости машин системы ЕС ЭВМ с системой*** IBM-360 рассматривалось как одно из основных мероприятий, способствующих распространению и росту в стране прикладного математического (программного) обеспечения. В целом общественное мнение, в том числе мнение ответственного разработчика проекта, склонялось к тому, что нужно взять за основу архитектуру IBM-360. Восьмибитный байт был главнейшим отличием архитектуры IBM 360, эффективно работать с ним не могла ни одна отечественная ЭВМ. Не принять его для машин «Ряда» означало крайне затруднить информационную совместимость с западными ЭВМ, что даже в условиях ***«железного занавеса»*** считалось нежелательным.

Было рекомендовано использовать как прототип, логическую структуру и систему команд, ***принятую в*** IBM-360. Это решение было принято практически при поддержке представителей организаций, которым предстояло работать по программе «Ряд». В результате решением МРП в феврале 1967-го года ***руководство разработкой аванпроекта поручило КБПА,*** известному созданием высокопроизводительных машин «Весна» и «Снег». Функции головной организации по математическим вопросам разработки «Ряда» должен был выполнять ИПМ АН СССР (Михаил Романович Шура-Бура).

В ***первой половине*** 1967-го года коллективом КБПА во главе с Владимиром Константиновичем Левиным был представлен «Аванпроект комплекса типовых информационно-вычислительных машин» (ОКР «Ряд») [11]. В нем предлагалась разработка по архитектуре IBM-360 четырех полностью совместимых моделей – производительностью 10–20, 100, 500 и 2000 тыс. операций в секунду. Во ***второй половине*** 1967-го года под руководством М.К. Сулима прошло обсуждение аванпроекта,

определение организаций-исполнителей работ, подготовка ***постановления правительства*** по дальнейшему развитию вычислительной техники [11]. Этим постановлением, вышедшим в ***декабре*** 1967-го года, разработка поручалась: Проектному бюро Минского завода им.

Г.К. Орджоникидзе, – Ереванскому НИИ математических машин, и – вновь создаваемому Научноисследовательскому центру электронной вычислительной техники (НИЦЭВТ). С начала 1968-го года развернулось проектирование машин во всех организациях, в том числе и в НИЦЭВТ.

Разработка операционных систем была в 1968 году поручена коллективам только что образованного НИЦЭВТа (основная операционная система ОС ЕС) и Минского проектного бюро завода им. Г.К. Орджоникидзе (дисковая операционная система ДОС ЕС). Научное руководство разработкой обеих систем в рамках Совета главных конструкторов (СГК) взял на себя профессор М.Р. Шура-Бура. Однако разработчики семейства Урал во главе с Б.И. Рамеевым, так же, как Виктор Михайлович Глушков, ***предлагали вести новую разработку на основе отечественного опыта*** с учетом зарубежных достижений. В ***октябре*** 1967 года они написали следующее письмо в Минрадиопром (приводятся фрагменты), которому была поручена правительством разработка ЕС ЭВМ [3]:

«Решение о разработке единого ряда электронных математических машин, предназначенных для использования в народном хозяйстве, правильное и своевременное. Оно призывает к объединению усилий коллективов разработчиков математических машин. Нужно ожидать, что это позволит резко увеличить производство математических машин, благодаря единой технологической и конструктивной основе, и даст возможность использовать единое математическое обеспечение для большинства применений.

Необходимо учитывать, что система IBM-360, являясь разработкой 1963-го – 64-го годов, уже в настоящий момент начинает отставать от уровня требований, предъявляемых к математическим машинам. Предложение о копировании системы IBM-360 эквивалентно планированию производства математических машин в семидесятые годы ***на уровне математических машин начала шестидесятых годов.*** Учитывая тенденцию развития науки и техники, можно смело утверждать, что в семидесятые годы архитектура системы 1BM-360 будет устаревшей, неспособной удовлетворить требования, предъявляемые к вычислительной технике. Архитектура системы IBM-360 имеет ряд недостатков, без устранения которых

недопустима разработка ряда машин, предназначенных для использования в ближайшее десятилетие, так как совокупность этих недостатков делает систему не соответствующей даже сегодняшним требованиям.

Копирование зарубежной разработки исключит возможность использования собственного опыта, накопленного коллективами разработчиков математических машин, и на ближайшие годы приведет к отказу от начала разработок, использующих новые принципы. Все это приведет к торможению развития вычислительной техники в стране. Коллективы разработчиков отечественных математических машин имеют достаточный опыт для разработки рядов машин, соответствующих уровню требований, которые будут предъявлены к вычислительной технике в ближайшие годы».

На ЭВМ семейства IBM-360, выпускаемых в те годы, сложные системы обработки информации и управления в реальном времени построить было невозможно. Они предназначались в основном для пакетной обработки данных в вычислительных центрах. Переход на интегральную элементную базу и дальнейшее развитие структуры и архитектуры «Урал», обеспечили бы возможность создания совершенной системы средств вычислительной техники.

Разработчики крупных государственных и военных систем вынуждены были бы использовать сложные ***прикладные программные компоненты и*** ОС IBM-360 ***низкого качества, для которых не было достоверной документации.*** Их ***нелегальные, неопределенные версии*** требовали виртуозного ***«раскапывания»*** логики и смысла процедур для установления причин проявления дефектов при попытках их применения в сложных проектах комплексов программ. Для успешного воспроизведения заимствованного программного обеспечения ***было необходимо***:

• иметь полный комплект документации по программному обеспечению системы-прототипа, достаточный для производства, сопровождения и эксплуатации;

• установить контакт с фирмой для сопровождения передаваемой информации и оказания помощи в использовании этой информации;

• информация по системе-прототипу должна была быть достаточной для обеспечения одинаковости программного обеспечения и функционирования средств ЕС ЭВМ и системы-прототипа;

• в распоряжении разработчиков программ должны были быть машины-прототипы, оснащенные полным, согласованным программным обеспечением, которое предполагается

воспроизвести.

Однако выбор в качестве прототипа системы IBM-360, ***исключал выполнение указанных выше условий.*** Фирма IBM не стремилась к сотрудничеству с Советским Союзом. Имевшаяся в Союзе документация по программному обеспечению системы IBM-360 была неполной, так как поступала не от фирмы-производителя, а от случайных зарубежных фирм. Закупка моделей системы IBM-360 была возможна лишь через страны-посредники, что создавало немалые трудности в освоении программных средств. Хотя для освоения, адаптации и документирования программных продуктов были выделены специальные предприятия (например, НИИ Интеграл, отделение в НИЦЭВТ), такие работы шли медленно и ***усугубляли отставание*** в эффективном применении поступавшей в 1970-е – 80-е годы аппаратуры ЕС ЭВМ.

С ***начала*** 1968 года к исследованиям, ведущимся в СССР по унифицированному ряду ЭВМ, ***стали проявлять интерес*** научные и промышленные организации ***стран социалистического содружества —*** СЭВ. После длительных консультаций, совещаний и согласований в начале 1969-го года было подписано многостороннее соглашение о сотрудничестве в области создания, производства и применения средств вычислительной техники. В межправительственном постановлении была поставлена задача разработать Единую Систему ЭВМ стран социалистического содружества (ЕС ЭВМ). Этим постановлением была образована Межправительственная комиссия по вычислительной технике (МПК по ВТ) на уровне министров, возглавляемая постоянным председателем – заместителем председателя Госплана СССР и ***генеральными конструкторами*** ЕС ЭВМ – С.А. Крутовских, (1968–1969 годы), А.М. Ларионовым (1970–1977 годы), В.В. Пржиялковским (1977–1990 годы), одновременно являвшихся в соответствующее время директорами НИЦЭВТ.

Западноевропейские фирмы, производящие вычислительную технику, желая быть конкурентоспособными с фирмой IBM, учитывая огромный научный и производственный потенциал Советского Союза, а также неудовлетворенный спрос на ЭВМ в СССР и странах Восточной Европы, первыми сделали конкретные шаги по установлению сотрудничества с Советским Союзом в области создания и производства вычислительной техники. ***Инициатором выступила крупнейшая английская фирма*** ICL, разработавшая к этому времени семейство ЭВМ Система-4, не уступающее IBM-360. Б.И. Рамеев был активным сторонником и участником переговоров. Он считал, что при тесном

сотрудничестве с фирмой ICL в соответствии с уже подписанными протоколами, Система-4 могла бы быть воспроизведена одним-двумя заводскими КБ. Основные силы НИИ и КБ страны можно было бы направить на создание более совершенного ряда машин на базе накопленного опыта с учетом новейших зарубежных достижений [3, 32, 34]. ˆ

На первой сессии Совета главных конструкторов в ***январе*** 1969-го года были утверждены все основополагающие решения, в том числе по архитектуре новой системы ЭВМ, в качестве которой принята архитектура IBM-360 [10]. Другим важным решением, принятым на первой сессии, было решение о ***контроле разработки военной приемкой министерства обороны СССР,*** а также о единой документации, согласованной с министерством обороны для всех отечественных ЭВМ Единой системы. ***Впервые в истории стран социалистического содружества началась реализация общего проекта,*** к разработке и производству этих машин было привлечено около 100 организаций и предприятий, более 200 тыс. ученых, инженеров и техников, около 300 тыс. рабочих из стран СЭВ.

В ***сентябре*** 1969-го года при приемке отечественной части технического проекта ЕС ЭВМ Государственной комиссией, заместитель генерального конструктора ЕС ЭВМ Б.И. Рамеев, ответственный за создание программного обеспечения, фактически ***снова поставил вопрос о переориентации ЕС ЭВМ*** с архитектуры IBM-360 на архитектуру системы Системы-4 и Сименс-4004. Б.И. Рамеева поддержал заместитель министра радиопромышленности М.К. Сулим. Против решительно выступили ИПМ (М.Р. Шура-Бура), ИНЭУМ (Б.Н. Наумов), а также Минский филиал НИЦЭВТ (В.В. Пржиялковский), НИИсчетмаш (В.Б. Ушаков) и генеральный конструктор С.А. Крутовских. В ***декабре*** 1969-го года министр радиопромышленности В.Д. Калмыков, рассмотрев проблему в присутствии М.В. Келдыша, М.Е. Раковского, А.А. Дородницына, С.А. Лебедева, М.Р. Шуры-Буры, С.А. Крутовских, ***принял решение*** продолжать работы по ранее согласованному направлению ***по архитектуре IBM-360.***

Срочную необходимость поправить положение в стране с программным обеспечением, подчеркивал председатель комиссии по вычислительной технике АН СССР и ГКНТ, академик А.А. Дородницын. В своем докладе на коллегии ГКНТ в ***сентябре*** 1969-го года он утверждал, что «по содержательной части математического обеспечения мы стоим на уровне, примерно, 1960 года, по сравнению с США». Этот доклад, отражавший действительное положение дел в стране, ***резко контрастировал с заявлениями некоторых популярных деятелей науки о***

превосходстве советской программистской школы над западной. В действительности, положительный отечественный опыт 1950 – х – 60-х годов в этой области относился в основном к ***академическому, индивидуальному программированию относительно небольших задач.*** Крупные программные комплексы были сосредоточены на оборонных предприятиях, не были востребованы и доступны для осознания и оценки в открытых научных и производственных организациях. Подобные проекты только начинали появляться на предприятиях народного хозяйства. Быстрый рост ресурсов ЭВМ общего назначения стимулировал естественное укрупнение проектов прикладных программных средств и необходимость перехода от ***«академического программирования»*** к коллективному созданию высококачественных программных продуктов с применением современных ***методов и средств программной инженерии.***

В 1971-м году прошли совместные испытания первой машины Единой системы – отечественной ЭВМ ЕС-1020, разработанной Минским НИИЭВМ. Информационная и программная совместимость с наиболее распространенными в мире ЭВМ, являвшимися дефакто мировыми стандартами, была достигнута ***в трудных условиях отсутствия документации и работающих образцов машин IBM-360. С*** 1972-го года началась поставка ЕС-1020 с операционной системой ДОС (см. главу 3). С 1973-го года поставлялась операционная система ОС ЕС, обеспечивавшая мультипрограммный режим.

К ***концу*** 1973-го года по программе ЕС ЭВМ прошли испытания шесть моделей. Параллельно в эти же годы были разработаны две версии ДОС и две версии ОС ЕС. Программа создания ЕС ЭВМ первой очереди была практически завершена. Выставка «ЕС ЭВМ-73», открывшаяся в июне 1973 года, подвела итог проделанной работе, показав возможности стран социалистического содружества при объединении усилий. Впервые ***новый ряд машин, получивший название ЕС ЭВМ-2 (Ряд-2),*** был обсужден на совещании главных конструкторов социалистических стран в июле 1972-го года, проходившем под председательством генерального конструктора

Александра Максимовича Ларионова. СГК развивал эту программу, и ***в*** 1973 году ***она, была утверждена.***

В 1975 – 1976-е годы ***состоялись контакты*** между руководством МРП СССР и НИЦЭВТ и представителями компании IBM [3, 23]. Первоначально со стороны компании IBM был проявлен интерес к сотрудничеству с МРП. Однако компания IBM не смогла добиться от правительства США согласия на сотрудничество с МРП, вяло текущие переговоры постепенно

затихали. ***Эмбарго на поставку в СССР вычислительных машин серьезно осложнило задачу обеспечения совместимости ЕС ЭВМ с машинами IBM.*** Тем не менее, к концу 1978-го года программа разработки ЕС ЭВМ-2 была практически завершена.

Для машин ЕС ЭВМ-2 были разработаны две новые оригинальные ОС: ДОС-3.1 и ОС 6.1. ***Выставка*** 1979-го года показала масштабы использования средств ЕС ЭВМ в народном хозяйстве. На это время ЕС ЭВМ занимала уже 72 % в общем парке ЭВМ страны. В стране выпускались 6 моделей ЭВМ и 42 типа периферийных устройств. Только в период 1975-е – 79-ые годы было задействовано более 700 административных, автоматизированных систем различного уровня, целиком построенных на технике ЕС ЭВМ. Наиболее крупные системы работали в Госплане, Госснабе, ЦСУ СССР, ГКНТ, Госстандарте и многих других ведомствах. Машинами ЕС ЭВМ пользовались крупнейшие институты АН СССР и высшие учебные заведения.

Особое место в программе развития машин ЕС ЭВМ-2 занимала модернизация ЭВМ ЕС-1060. С 1983 по 1988 год было продано 566 машин. К ***началу*** 80– х ***годов машины ЕС ЭВМ практически удовлетворили спрос стран СЭВ в машинах общего назначения.*** К ***середине*** 1984-го года появилась первая очередь машин ЕС ЭВМ-3. Отечественные ЭВМ ЕС-1036, 1046 и 1066 ***снабжались оригинальной операционной системой ОС-7,*** состоящей из системы виртуальных машин (СВМ) и базовой операционной системы (БОС) (см. главу 3). Многие пользователи ЕС ЭВМ, в первую очередь такие крупные, как ЦК КПСС, СМ СССР, КГБ СССР, некоторые министерства, использовали ***оригинальные операционные системы IBM – VM и MVS.*** Около 20 % от выпуска поставлялось министерству обороны, шел устойчивый экспорт машин в страны содружества и страны третьего мира – Индию, Вьетнам, Китай, Кубу, страны Ближнего Востока.

В ***мае*** 1987-го года была одобрена концепция создания ***ЕС ЭВМ Ряд-4. Постановлением правительства*** была утверждена отечественная часть программы создания технических и программных средств ЕС ЭВМ-4, предназначавшихся «для решения широкого круга задач в вычислительных сетях и центрах коллективного пользования, АСУ различного уровня, АСПИ и САПР, с технико-экономическими показателями ***на уровне мировых достижений.*** Использование зарубежной концепции, архитектуры и операционных систем (ОС) IBM-360 ***позволило освоить и творчески применить их как фундамента при промышленной разработке мощного унифицированного*** семейства ***ЕС ЭВМ*** для отечественного народного хозяйства и оборонных систем. Распыленные по многим типам ЭВМ средства

и силы наших специалистов промышленности смогли быть сконцентрированы на создании и развитии единой системы компонентов вычислительной техники и инструментальных операционных систем. Такая концентрация и координация усилий, разрозненных академических и отраслевых организаций со своими интересами и планами, была невозможной без жесткого планирования, высокой дисциплины и ответственности за выполнение заданий. Кроме того, учитывалось, что к 1985 году истечет 10-летний срок эксплуатации: ***5500 различных ЭВМ общего назначения,*** в том числе всех машин Урал-11, 14, 16 (325 машин), БЭСМ-4 и БЭСМ-4м (441 шт.), М-220 и М-222 (502 шт.). Прогнозировался вывод из эксплуатации: 195 ЭВМ БЭСМ-6, из состава выпущенных на тот период 355 машин. Таким образом, машины ЕС ЭВМ должны были к этому времени составить практически 100 % парка универсальных ЭВМ общего назначения [23].

В ***январе*** 1986-го года вышло ***постановление правительства СССР*** по организации производства персональных ЭВМ (ПЭВМ) и предписывалось МРП, МЭП и Минприбору в короткие сроки освоить выпуск ПЭВМ, совместимых с IBM PC, в количестве около 1 млн. шт. в год. Разработка ПЭВМ в МРП поручалась Минскому НИИЭВМ. В короткие сроки НИИЭВМ разработал 12 типов ПЭВМ. Однако с каждым годом сокращалось финансирование, нарастало ***смутное время,*** работы по производству и развитию отечественной вычислительной техники ***затихли.***

2.5. История семейства малых универсальных ЭВМ в 1970-е – 80-е годы

Основная задача создания системы малых ЭВМ – обеспечить возможности широкого использования средств вычислительной техники, баз данных и баз знаний в большинстве сфер ***приложения человеческого труда,*** не накладывая никаких специальных требований на выбор архитектуры основных схемотехнических компонентов ***управляющих ЭВМ,*** какими являются малые и микроЭВМ. Ориентация на конкретные применения достигалась путем использования специализированных периферийных устройств и развитого набора средств программирования. В середине семидесятых годов по единой для стран – членов СЭВ ***долгосрочной целевой программе*** была начата разработка ряда моделей ***управляющих малых ЭВМ массового применения.*** Было развернуто крупносерийное производство Системы малых и микроЭВМ (СМ ЭВМ). Тысячи таких ЭВМ установлены на предприятиях и в научно-исследовательских организациях; они использовались в автоматизированных комплексах промышленного назначения, в составе сложных экспериментальных установок, в здравоохранении, связи, сфере услуг, на транспорте.

Опыт применения малых и микроЭВМ позволил в полной мере выявить круг задач, которые необходимо решать в процессе внедрения интегрированных систем автоматизированного проектирования и производства. В массовых областях применения вычислительной техники малые и микроЭВМ призваны были сыграть основную роль в информатики и теории управления, а также в методах машинной графики, цифровой обработки сигналов, анализа и синтеза речевых сообщений, построения сетей передачи информации взаимосвязанных объектов управления.

В 1974-м году решением ***Межправительственной комиссии по сотрудничеству социалистических стран в области вычислительной техники*** (МПК по ВТ), Институт электронных управляющих машин (ИНЭУМ) [27] был определен ***головной организацией по Системе малых (СМ) ЭВМ,*** а Борис Николаевич Наумов – руководителем работ. Важные особенности проекта состояли в широкой номенклатуре конкретных семейств ЭВМ и инструментальных технологических программ, обеспечивавших ***возможность пользователям формировать наборы средств автоматизации программирования*** для различных сфер

применения. Комплексом научно-исследовательских и опытно-конструкторских работ по СМ ЭВМ занималось более 30 институтов и предприятий СЭВ. Эти средства развивались с ***середины*** 70-х годов, а основные достижения и внедрение системы малых ЭВМ относятся к 80-м годам. Эти семейства машин обычно производились большими тиражами (тысячи экземпляров), в каждой из которых комплексы прикладных программ для конкретного применения были среднего размера (десятки тысяч строк) и разрабатывались небольшими коллективами. При разработке СМ ЭВМ было принято ***несколько общих принципов:***

обеспечение преемственности в прикладном программном обеспечении по отношению к ЭВМ и УВК, выпускавшимся в стране ранее, – моделям АСВТ-М: М-400 (СМ3, СМ4, СМ-1300, СМ-1420), М-5000 (СМ-1600), М– 6000/7000 (СМ1, СМ2, СМ-1210);

• построение систем с разделением функций, использующих универсальные и специализированные процессоры СМ ЭВМ;

• развитая номенклатура адаптеров передачи данных для сопряжения СМ ЭВМ с линиями связи в соответствии с международными стандартами;

• наличие средств сопряжения СМ ЭВМ с ЕС ЭВМ в гетерогенных системах;

• построение проблемно-ориентированных комплексов, выпускаемых промышленностью на базе моделей СМ ЭВМ: специфицированные управляющие вычислительные комплексы (УВК), поставляемые заводами по спецификациям заказчиков; измерительно-вычислительные комплексы (ИВК) с аппаратурой САМАС; автоматизированные рабочие места (АРМ) для САПР в машиностроении, радиоэлектронике и строитель ств е.

Гибкость и модульность средств СМ ЭВМ, наличие развитых средств сопряжения с ЭВМ при применении, наличие проблемно-ориентированных системных и прикладных программных средств СМ ЭВМ обеспечили широкое использование ИВК в промышленности и системах автоматизации научных исследований. В состав АРМов входил широкий набор базового программного обеспечения машинной графики. Наибольшее применение нашли АРМы, для радиоэлектроники, машиностроения, строительного проектирования, обработки экономической информации. В составе СМ ЭВМ было создано ***несколько семейств микро– и мини-ЭВМ, управляющих и вычислительных комплексов на базе этих ЭВМ.***

Семейство УВК СМ1, СМ2, СМ-1210 класса 16-разрядных мини-ЭВМ обладало полной программной совместимостью с

М-7000 и односторонней совместимостью на уровне перемещаемых программ с М-6000. Программное обеспечение УВК СМ1 и СМ2 было построено по модульному принципу, что позволяло компоновать программные системы в соответствии с требуемыми режимами работы и выполняемыми функциями на заданной конфигурации технических средств. В ***составе технологического программного обеспечения*** были предусмотрены:

• многозадачная однопроцессорная ОС, обеспечивавшая приоритетную организацию выполнения задач и защиту памяти;

• многозадачная мультипроцессорная операционная система для УВК СМ2, обеспечивавшая выполнение на двух процессорах двух старших по приоритету задач;

• операционные системы М-6000, адаптированные к однопроцессорным конфигурациям СМ1 и СМ2 с объемом оперативной памяти не более 32 К слов;

• библиотеки подпрограмм;

• проблемно-ориентированный пакет макроопределений, позволяющий проектировщику АСУТП компоновать системы сбора, анализа и обработки технологической информации;

• система подготовки прикладных программ на мнемокодах М-6000 и М-7000, макроязыке СМ1 и СМ2 (уровня макроассемблера), языках Фортран-IV, диалекте Алгол-60 и языке Бейсик.

Разработка этого семейства была выполнена НПО «Импульс» (г. Северодонецк) под руководством В.В. Резанова, В.М. Костелянского. Серийный выпуск освоили Северодонецкий приборостроительный завод и «Орловский завод УВМ им. К.Н. Руднева» На объекты было поставлено около 17 тыс. УВК СМ1, СМ2, СМ-1210, в том числе более 10 тыс. для систем управления процессами. Наиболее широко они использовались в системах энергетического и военного назначения.

Семейство УВК СМ3, СМ4, СМ-1420, СМ-1425 класса 16-разрядных мини-ЭВМ обладало программной совместимостью с М-400 и семейством PDP-11 фирмы Digital Equipment. В ***состав поставляемого программного обеспечения*** семейства были включены:

• дисковая и резидентная (в оперативной памяти) операционные системы);

• семейство совместимых мультипрограммных операционных систем реального времени (ОС РВ) с большим числом уровней приоритета для различных конфигураций технических средств (СМ4, СМ-1420, СМ-1425);

• дисковая диалоговая многотерминальная ОС с разделением

времени ДИАМС;

• однопользовательская дисковая фоново оперативная базовая операционная система ФОБОС;

• инструментальная мобильная операционная система ИНМОС типа Unix;

• пакеты программ обработки графической информации;

• системы программирования, включающие трансляторы с языков: ассемблер, макроассемблер, Фортран-IV, Бейсик и диалоговый язык ДС СМ;

• проблемно-ориентированные пакеты прикладных программ, в том числе для управления лабораторными экспериментами, использования в медицине для обработки данных экономического характера.

Серийное производство было освоено московским заводом Энергоприбор и Киевским заводом ВУМ [27]. Семейства СМ3, СМ4, СМ-1420, СМ-1425 ***не были копиями зарубежных прототипов,*** но обеспечивали программную совместимость с семейством мини-ЭВМ, в то время наиболее распространенным на Западе. При разработке моделей Единой Системы и СМ ЭВМ была поставлена ***цель в максимальной мере обеспечить их совместимость с ЭВМ, разработанными в других странах.*** Такая цель была вполне оправданна, поскольку в противном случае отечественная вычислительная техника была бы изолирована от мировых достижений в области компьютерной технологии и, в частности, принципиально не имела бы доступа к накопленному в мире программному продукту.

Семейство 32-разрядных вычислительных комплексов СМ-1700 разрабатывалось под руководством Н.Л. Прохорова, генерального конструктора СМ ЭВМ с 1984 года. Оно обладало программной совместимостью с семейством VAX-11 фирмы Digital Equipment и односторонней совместимостью с 16-разрядными моделями семейства СМ3, СМ4, СМ-1420, СМ-1425. Архитектура СМ-1700 поддерживала организацию виртуальной памяти, реализуемую с помощью контроллера управления памятью.

Семейство УВК СМ-1800 на базе микро-ЭВМ представляло собой 8-разрядные микро-ЭВМ на базе микропроцессора КР580, построенные по магистрально-модульному принципу с системным интерфейсом И41 (Multibus), принятым в качестве стандарта СМ ЭВМ. Модель СМ-1804 представляла собой вариант СМ-1800 в промышленном исполнении для использования на предприятиях с ограниченным доступом обслуживающего персонала [11].

В 1981-е – 90-е годы было выпущено более 11 тыс. УВК СМ-1800, СМ-1803, СМ-1804, а в 1987 – 90-е годы – более 18 тыс.

УВК СМ-1810, СМ-1814, СМ-1820. Принципы технологии и стандарты СМ ЭВМ охватывали все аспекты унификации элементов, устройств моделей ЭВМ и комплексов на их основе, а также технологических программных средств.

2.6. История мобильных и бортовых специализированных ЭВМ в 1960-е – 80-е годы

Во многих организациях оборонной промышленности в конце 50-х годов начали разрабатываться многочисленные ***оригинальные специализированные ЭВМ, в которых отсутствовало копирование зарубежных аналогов.*** При создании требований к таким объектным ЭВМ военного назначения для эффективного использования их ограниченных вычислительных ресурсов необходим был детальный анализ алгоритмов и программ, подлежащих реализации. Относительно узкая ориентировка каждого типа ЭВМ на совершенно определенные задачи открывала возможность значительной экономии оборудования и улучшения характеристик по памяти и производительности на имеющейся элементной базе низкого качества. С другой стороны, в то время от программистов требовалась максимальная эффективность использования ограниченных доступных ресурсов и знаний тонкостей архитектуры объектных ЭВМ при реализации алгоритмов, что, в частности, определило широкое применение машинно-ориентированных языков программирования – автокодов.

В 1950-е – 60-е годы развитие в стране технологии производства и элементной базы специализированных ЭВМ ***не поспевало за ростом требований*** к их ресурсам по памяти и производительности, необходимым для реализации новых расширяющихся задач заказчиков систем в оборонной промышленности. Очень быстро увеличивалась сложность и ответственность задач обработки информации и управления, возлагаемых на ЭВМ, что вызывало рост требований к качеству, надежности функционирования и безопасности применения комплексов программ для военных систем реального времени. Для обеспечения решения этих сложных задач в очень ограниченных вычислительных ресурсах, ***архитектура и системы команд, специализированных ЭВМ должны были тщательно адаптироваться*** к характеристикам прикладных задач и сфер применения систем военного назначения [3, 16].

Одновременно и независимо многие подобные проблемы решались на ряде оборонных предприятий при создании ***мобильных вычислительных средств и комплексов программ*** для авиационных, морских, космических и других систем военного назначения. Особенности функциональных задач и

требований сфер применения, а также жесткие межведомственные барьеры и ***ограничения по секретности*** привели к тому, что обмен информацией о методах, свойствах и достижениях при разработках, ***специализированных ЭВМ и крупных программных продуктов*** между специалистами разных оборонных отраслей, и предприятий в 60-е годы в стране был резко ограничен. Также почти ***отсутствовала информация о технических характеристиках и принципиальных особенностях вы числительных машин этого класса и программных средств за рубежом.***

Во второй половине 50-х годов в Ленинграде коллективом, руководимым Ф.Г. Старосом и И.В. Бергом, начали разрабатываться ***первые в стране мобильные, управляющие, полупроводниковые ЭВМ*** [3,11]. Особенностью этих в то время ***особо секретных*** работ была изначальная ориентация на микроэлектронные технологии. Это позволило получить первые в СССР крупные результаты в создании и внедрении образцов микроэлектронной управляющей вычислительной техники. В 1956-м году была организована специальная (секретная) лаборатория СЛ-11. Уже в первые годы ее существования были достигнуты серьезные результаты по созданию экспериментальных образцов пленочных микросхем, интегральных, ферритовых пластин для запоминающих устройств и логических узлов ЭВМ с малым потреблением энергии. В 1961-м году ***правительством было принято решение*** об организации самостоятельного КБ-2 электронной техники для оборонных систем под руководством Ф.Г. Староса [11].

Первым крупным результатом этой организации, выполненным в два года, явилась разработка бортовой, управляющей ЭВМ УМ1-НХ. В 1962-м году она была принята Государственной комиссией под председательством академика А.А. Дородницына и рекомендована к серийному производству. ЭВМ УМ1-НХ стала предвестницей появления нового класса вычислительной техники ***мобильных, микроэлектронных управляющих ЭВМ.*** Существенными отличительными характеристиками УМ1-НХ явились низкая для того времени стоимость и высокая надежность работы машины в производственных условиях. По постановлению правительства в 1963-м году началось освоение и серийное производство УМ1-НХ на Ленинградском электромеханическом заводе (ЛЭМЗ). В последующие годы ЛЭМЗом было освоено производство новых устройств для УМ1-НХ, расширяющих её возможности. Используя их вместе с базовым конструктивом УМ1-НХ, завод выполнял заказы промышленности на управляющие комплексы для

конкретных оборонных систем. В 1964-м году в КБ-2 под руководством Ф.Г. Староса была разработана микроминиатюрная ЭВМ УМ-2, ориентированная на применение в аэрокосмических и авиационных системах. Кроме достаточно развитой архитектуры, УМ-2 имела оригинальные конструктивные и технологические решения, которые оказали большое влияние на развитие бортовой вычислительной техники в последующие годы.

Разработка УМ-2, ее удачные архитектурные и конструктивно-технологические решения получили свое развитие и практическое внедрение по двум направлениям: была разработана управляющая ЭВМ «Электроника К-200» и управляющий комплекс с наращиваемыми устройствами ввода-вывода и периферийными устройствами, получивший название «Электроника К-201». Разработки больших интегральных схем послужили базой для развития работ по созданию ***мобильной машины «Электроника С5» —*** первого в СССР семейства одноплатных, многоплатных и однокристальной микро-ЭВМ для управления объектами и процессами в реальном времени. Среди этого семейства с оригинальной структурой и архитектурой, в разработке которых приняли участие ученые Института кибернетики АН Украины, следует особо выделить однокристальную микро-ЭВМ «С5-31». [11].

В начале 60-х годов особенно активно развивались ***мобильные ЭВМ*** для систем по заказам министерства обороны СССР. Для системы ПВО в Морском научно-исследовательском институте (МНИИ-1) была создана мобильная (в кузовах) полупроводниковая ЭВМ 5Э89 (Главный конструктор – Я.А. Хетагуров) [28]. Разработка завершилась в 1962 году приемными испытаниями и передачей ЭВМ в серийное производство. Ее память составляла 14 тысяч команд, и две тысячи слов оперативной памяти, быстродействие около 60 тыс. операций в секунду. На машине выполнялась обработка информации, поступающей в реальном масштабе времени от радиолокаторов кругового обзора, и автоматизированное сопровождение воздушных целей, истребителей и ракет. Для повышения надежности и производительности была предусмотрена возможность совместной работы двух ЭВМ. До 1970 года ее выпускали на Ульяновском заводе им. Володарского, а затем на Загорском электромеханическом заводе, всего было поставлено в войска около 400 комплектов. В 80-е годы машина прошла модернизацию на новой элементной базе с уменьшением габаритов и энергопотребления, с полным сохранением логики команд ЭВМ 5Э89 и всего комплекса программ РЛУ. Производство модернизированной машины прекратилось только

в 1992 году.

В 1969-м году началась разработка ЭВМ 5Э26 для ЗРК С-300 (Главные конструкторы – С.А. Лебедев, В.С. Бурцев) [11]. В 1975-м году началось серийное производство этой мобильной, управляющей, многопроцессорной, высокопроизводительной ЭВМ, ***которая применялась в ряде оборонных систем.*** Она была построена по модульному принципу, с высокоэффективной системой автоматического резервирования, базирующейся на аппаратном контроле и обеспечивающей возможность восстановления процесса управления при сбоях и отказах аппаратуры. Работает в широком диапазоне климатических и механических воздействий, с развитым математическим обеспечением и системой автоматизации программирования.

Технические характеристики: производительность 1,5 млн. операций в секунду, длина слова 32 разряда, представление информации естественное, целое слово, полуслово, байт, бит, объем оперативной памяти 32–34 кбайт, объем командной памяти 64-256 кбайт. Независимый процессор ввода-вывода информации по 12 каналам связи – максимальный темп обмена свыше 1 Мбайт/с. Объем 2,5–4,5 м^3, потребляемая мощность 5–7 кВт. Выпускалась в двух модификациях, различающихся объемом памяти.

Принципиальные особенности ЭВМ:

впервые в СССР создана мобильная многопроцессорная высокопроизводительная структура с модульной памятью, легко адаптируемая к различным требованиям по производительности и памяти в различных системах управления;

• машина с автоматическим резервированием на уровне модулей и обеспечивающая восстановление вычислительного процесса при сбоях и отказах аппаратуры в системах управления, работающая в реальном времени;

• мобильная машина, снабжена развитым математическим обеспечением, эффективной системой автоматизации программирования и возможностью работы с языками высокого уровня;

• энергонезависимая память команд на микробиаксах с возможностью электрической перезаписи информации внешней аппаратурой записи;

• введена эффективная система эксплуатации с двухуровневой локализацией неисправной ячейки, обеспечивающая эффективность восстановления аппаратуры среднетехническим персоналом.

В 1988-м году линию мобильных, управляющих, многопроцессорных вычислительных систем, начатую ЭВМ 5Э26,

продолжила машина 40У6 (серийное производство). Главный конструктор Е.А. Кривошеев) [11]. Эта ЭВМ построена по модульному принципу, с высокой жизнеспособностью за счет дублирования некоторых модулей и резервирования, базирующейся на мощной системе аппаратного контроля и обеспечивающей возможность восстановления процесса управления при сбоях и отказах аппаратуры. ЭВМ работает в режиме жесткого реального времени, рассчитана на работу в широком диапазоне климатических и механических воздействий, имеет развитое обеспечение автоматизации программирования.

Технические характеристики. 32-разрядное слово, плавающая запятая, оперативная память 256 Кб (дублируется), командная память 512 Кб (дублируется), 15-канальный процессор ввода-вывода информации (дублируется). Процессор ввода-вывода имеет 13 специализированных каналов и 2 стандартных канала. Память команд 512 Кб, имеет внутренний контроль по кодам Хемминга, байтовый контроль передач, информация не пропадает от выключения питания, что обеспечивается переходом на аккумуляторное питание. Элементная база Маломощная серия ТТЛ-микросхем, КМОП-микросхемы памяти. Потребляемая мощность 5,5 кВт, объем 2,5–4,5 куб. м.

Программное обеспечение. Трансляторы с автокода, Фортрана, СИ, Паскаль.

Разработка средств цифровой вычислительной техники ***для бортового оборудования самолетов, ракет, космических аппаратов*** началась в СССР во второй половине 60-х годов. В это время подавляющее большинство НИИ и приборостроительных КБ минавиа-прома (ОКБ «Электроавтоматика» – ЛНПОЭА, МИЭА, ГосНИИАС, НИИП) и минрадиопрома (НИИ «Аргон» – НИЦЭВТ, ВНИИРА, НПО «Вега», МНИИ «Агат») (везде использованы современные названия) и ряд других предприятий начали разработку макетов различного рода цифровых вычислительных устройств (ЦВУ), бортового оборудования самолетов, а затем ракет и космических аппаратов.

Цифровые вычислительные средства в составе бортового оборудования самолетов появились в начале 60-х годов и за относительно короткий срок практически полностью заменили используемые ранее аналоговые вычислители, поскольку обеспечивали более высокую точность решения задач, характеризовались большей универсальностью применения и обладали широкими логическими возможностями. Эти качества бортовой цифровой вычислительной машины (БЦВМ) позволяют использовать ее практически во всех подсистемах бортового оборудования самолета, обеспечивают устойчивость БЦВМ к

усложнению алгоритмов и позволяют применять более сложные, а значит, и более совершенные алгоритмов управления самолетом и его подсистемами. Они позволили осуществить информационное взаимодействие между отдельными (ранее непосредственно не взаимодействовавшими) подсистемами бортового оборудования и образовать единый комплекс бортового оборудования (КБО), что, в конечном счете, повысило эффективность выполнения полетного задания и безопасность полета.

Решением ***комиссии Президиума СМ СССР по военно-промышленным вопросам*** в 1963-м году Научно-исследовательский институт электронных математических машин (НИЭМ) был назначен головным предприятием страны по бортовым ЭВМ [11], а в 1986-м году выделилось в самостоятельное предприятие – НИИ «Аргон». За это время ***было разработано более 30 типов БЭВМ*** и вычислительных комплексов на их основе. Создание ряда «Аргон» шло в три этапа. На первом этапе (1964-й год – ***середина*** 70-х годов) были разработаны ***11 моделей машин*** для ракетно-космических, авиационных и наземных автоматизированных систем управления. Базой для первых моделей послужил созданный к этому времени научно-технический задел по ЭВМ общего назначения.

К ЭВМ, используемым в составе систем управления летательных аппаратов, предъявлялся ряд специфических требований, которые значительно усложняли проектирование бортовых машин. К числу важнейших требований, во многом определявших выбор основных проектных решений, относились ограничения на массогабаритные характеристики и потребляемую мощность, необходимость придания повышенной надежности функционирования, устойчивости к широкому диапазону внешних воздействий (механических, климатических, радиационных), возможность обмена в реальном времени информацией с разнообразными датчиками и исполнительными устройствами систем управления.

Создание БЭВМ на. первом этапе велось на основе ряда принципиальных положений, выработанных с учетом специфических требований к бортовым машинам в результате многочисленных исследований, осуществления эскизных и технических проектов. С самого начала было принято решение проектировать БЭВМ на новой для того времени элементной базе – ***интегральных схемах*** (ИС). Только применение ИС давало возможность обеспечить необходимые параметры машин, в первую очередь массогабаритные, энергетические и прочностные.

Работы по созданию ряда «Аргон» дали мощный толчок развитию элементной базы для ЭВМ оборонного значения. НИЭМ и его преемники, были инициаторами, заказчиками и соисполнителями разработки ряда выпускавшихся крупными сериями ИС. Некоторые из них получили широкое применение не только в бортовых, но и в стационарных ЭВМ.

Введение ***программной совместимости*** между моделями БЭВМ в то время ***было признано нецелесообразным.*** Это потребовало бы разработки единой для всех машин сложной системы команд, часть из которых во многих случаях оказалась бы излишней. При достигнутом в тот период уровне технологии производства элементов единственным путем удовлетворения разнообразных, жестких требований к бортовым ЭВМ была ***специализация систем команд к решаемым задачам.*** Вместе с тем системы команд и организация вычислений для различных моделей строились на основе общих исходных принципов и являлись достаточно близкими. Для повышения плотности компоновки ИС использовался многослойный печатный монтаж. С целью сокращения сроков разработки и числа ошибок при выполнении ручных операций, основные узлы БЭВМ создавались с применением систем автоматизированного проектирования на базе универсальных ЭВМ.

Машины первого этапа по типу используемой элементной базы разделялись на ***две группы.*** К ***первой,*** более ранней по времени разработки группе, относились БЭВМ «Аргон-1», «Аргон-10», «Аргон-11А», «Аргон-12А», собранные на гибридных ИС типа «Тропа». Серия гибридных ИС «Тропа-1» была предложена НИЭМ совместно с НИИТТ министерства электронной промышленности для первых БЭВМ ряда «Аргон». ***Вторую группу*** составили – БЭВМ «Аргон», выполненные на твердотельных ИС, а «Аргон-17» – на первых микропроцессорных БИС. БЭВМ ракетно-космического назначения в большинстве случаев являлись необслуживаемыми и строились по моноблочному принципу.

Серьезную проблему при проектировании бортовых ЭВМ представляло обеспечение ***устойчивости к негативному воздействию внешней среды.*** Ввиду их компактности существенное значение имел отвод тепла. В большинстве машин обеих групп было применено принудительное воздушное охлаждение их внутренних частей. Ракетные машины были помещены в герметизированный корпус, служащий радиатором для отвода тепла в окружающую среду.

Впервые ***резервирование аппаратуры*** было использовано в БЭВМ «Аргон 11С» – первой отечественной машине космического

назначения, осуществлявшей автоматическое управление полетом космического аппарата, совершившего облет Луны с возвращением спускаемого аппарата на Землю (программа «Зонд»). В ходе исследований, выполненных при проектировании этой машины, оптимальной структурой резервирования, обеспечивающей экономию машинных ресурсов и приемлемый уровень надежности, была признана ***троированная структура*** с голосованием по большинству (мажорирование). Эта структура получила дальнейшее развитие в бортовом вычислительном комплексе «Аргон-16» – ***уникальной разработке в мировой практике*** создания бортовых ЭВМ. За четверть века эксплуатации на космических кораблях Союз, транспортных кораблях Прогресс, орбитальных станциях Салют, Алмаз, Мир не было отмечено ***ни одного отказа комплекса в составе системы управления.*** За это время было выпущено более 300 экземпляров – ***рекордный показатель для машин космического применения.*** Высоконадежная троированная структура в модифицированном виде использована в БЭВМ, предназначенной для применения в инерциальной системе управления ракетой комплекса ПРО. Отличительная особенность «Аргон-17» – высокая радиационная стойкость аппаратуры, гарантирующая выполнение задачи в условиях воздействия ядерного взрыва. Работы первого этапа сыграли ***исключительно важную роль в развитии отечественной бортовой вычислительной техники.*** При разработке функциональных комплексов программ оборонных систем ***для ряда типов ЭВМ «Аргон» в 1970 – 80-е*** годы активно использовалась инструментальная ***система автоматизации программирования и отладки Яуза-6,*** адаптированная для соответствующих ЭВМ (см. главу 3) [11, 18].

Многие системы военного назначения на базе специализированных ЭВМ предполагались для производства и применения в относительно небольшом количестве (единицы, десятки, или сотни экземпляров), что ориентировало разработчиков на применение оригинальных технических решений и вызывало ***пренебрежение унификацией и стандартизацией*** аппаратуры, комплексов программ и технологий их производства. Практически независимая разработка такого широкого спектра вычислительных машин и комплексов программ, конечно, обходилась очень дорого, однако в результате появлялось множество эффективных технических решений при разработке и применении ЭВМ и комплексов программ. Ведомства заказчиков систем не координировали между собой технические требования к вычислительным средствам, каждое из которых вынуждено, адаптировалось

разработчиками к задачам конкретного заказчика.

Использование БЦВМ потребовало определенной унификации радиоэлектронного оборудования самолетов, в результате которой сократились сроки и снизились затраты на разработку и последующую модернизацию. Поэтому на ранних стадиях развития цифровой авионики основное внимание уделялось разработке БЦВМ и средств ее сопряжения с бортовой аппаратурой. Проблема создания программных продуктов обострялась по мере усложнения структуры машин, расширения круга решаемых задач, появления и развития бортовых вычислительных систем.

На основе анализа, проведенного в конце 1970-х – начале 80-х годов, была разработана программа создания семейств, унифицированных БЦВМ для использования на подвижных объектах всех классов. Эта программа была утверждена в 1984-м году решением Государственной Комиссии. В соответствии с ней были начаты работы по созданию унифицированных семейства БЭВМ – СБ3541 и СБ3542 с архитектурой типа «Электроника-32», а в НИИ «Аргон» – СБ5140 с архитектурой «ПОИСК».

На протяжении трех десятилетий БЦВМ качественно изменялись. Их быстродействие увеличилось более чем на три порядка и достигло десятков миллионов операций в секунду, а емкость запоминающих устройств достигает 8 -16 Мб. Одновременно уменьшились вес и энергопотребление. Это было обусловлено совершенствованием элементной базы, архитектуры и структурной организации машин. Замена дискретных компонентов большими интегральными схемами позволила повысить быстродействие машины более чем на два порядка при одновременном снижении на порядок и более энергопотребления и веса. Совершенствование микропроцессоров в 80-х и начале 90 – х годов позволило поднять еще как минимум на порядок быстродействие БЦВМ, также улучшились внутренние и внешние интерфейсы вычислительных машин.

В структуре ***БЦВМ второго поколения*** (начало 70-х годов) начинают использоваться элементы конвейеризации, обеспечивающие совмещение в выполнении операций, процессоры, содержащие более совершенные сумматоры и специальные устройства для выполнения операций умножения, деления и вычисления элементарных функций. Структура машин реализуется на интегральных схемах, но остается детерминированной и трудно модернизируемой, т. е., по существу, закрытой. Для написания программ начинают использоваться ***языки уровня ассемблера, а для их отработки – специальные отладочные комплексы,*** объединяющие БЦВМ с

инструментальной вычислительной машиной. К середине 80-х годов было разработано четыре модификации машины: «Аргон-15» (ОЗУ – 1К, ПЗУ – 24К слов) имеет массу 35 кг и наработку на отказ 500 ч. Быстродействие машины «Аргон-15К» – 500 тысяч, а «Аргон-15-М» – 800 тысяч коротких операций в секунду; имеет наработку на отказ 5000 ч и весит 16,6 кг.

В ***БЦВМ третьего поколения*** использованы иерархическая память, включающая сверхоперативную память (РОН, регистры общего назначения), многоуровневая система прерывания, каналы прямого доступа к памяти, а также механизмы защиты информации от несанкционированного доступа. В структуре бортовых машин третьего поколения начинают применяться средства поддержки мультипроцессирования. Структура этих БЦВМ имеет в основном магистрально-модульную организацию и допускает изменение характеристик машины в достаточно широких пределах путем использования необходимого количества соответствующих (унифицированных) модулей, т. е. приобретает некоторую открытость.

2.7. История семейства «Аргон» с архитектурой и системами команд ЕС ЭВМ в 1970-е – 80-е годы

Перед разработчиками ***специализированных, мобильных ЭВМ оборонного назначения*** в НИЦЭВТе в середине ***70-х годов*** были поставлены качественно новые задачи. Возникла необходимость внедрения бортовых ЭВМ, показавших высокую эффективность при управлении техническими средствами, в автоматизированные системы управления войсками, авиационные комплексы радиолокационного дозора и наведения, системы управления воздушным движением. По требуемым параметрам ЭВМ, предназначенные для работы в таких системах, были ***близки к стационарным универсальным машинам ЕС ЭВМ*** (решали преимущественно расчетные и информационные задачи, должны были иметь 32-разрядную сетку, высокую производительность, оперативную и внешнюю память большой емкости, оснащаться сложным программным обеспечением) [11].

В области управляющих ЭВМ под флагом унификации в 80-е годы в НИЦЭВТ велись работы по созданию бортовых машин ***с архитектурой и системами команд, ЕС ЭВМ.*** Это позволяло использовать для разработки управляющих программ реального времени стационарные ЕС ЭВМ без применения интерпретаторов и кросс-систем. Ориентировка этих машин на вычислительные задачи приводила к неэффективному их использованию при решении преимущественно логических задач, характерных для бортовых систем управления в реальном времени. Такие машины под маркой А-30, А-50 были использованы в некоторых системах военного назначения. Однако широкий парк, до тех пор применяемых, специализированных машин (серии «Аргон», 5Э26, 40У6) был ориентирован на особенности функциональных задач, и огромный объем испытанных, эксплуатируемых высококачественных прикладных комплексов программ поддерживал актуальность применения ***для разработки и развития программ реального времени, технологических кросс-систем, в том числе ЯУЗА-6 и РУЗА*** [11, 18].

К этому времени резко расширился парк эксплуатируемых БЭВМ, значительно возросли трудоемкость и стоимость их разработки. Отечественными предприятиями было создано большое число машин, предназначенных, как правило, для одной конкретной системы вооружения. Незначительно отличаясь по функциональным возможностям, они имели оригинальные

системы команд, структуру, конструктивные решения. По этой причине актуальность приобрела ***проблема унификации создаваемых моделей*** БЭВМ. Решение этой задачи стало возможным на пути перехода от отдельных моделей с несовместимыми системами команд к семействам ***машин единой архитектуры.***

Базовой архитектурой нового поколения БЭВМ, предназначенных для решения расчетных и информационно-логических задач с большими объемами обрабатываемой и хранимой информации, ***была выбрана архитектура стационарных машин ЕС ЭВМ на основе IBM-360,*** которая к этому времени утвердилась в качестве магистрального направления развития отечественных стационарных ЭВМ общего назначения. Для построения ряда перспективных БЭВМ важное значение имели свойства присущие ЕС ЭВМ: система программного обеспечения, универсальный набор команд, 32-разрядное слово, модульность, стандартизованные интерфейсы, мультисистемные свойства, наращиваемость функциональных возможностей. Совместимость с ЕС ЭВМ позволяла использовать серийные стационарные машины в качестве промежуточного ***стендового варианта на весь период отработки системы*** управления и тем самым ускорять создание БЭВМ, их программного обеспечения и системы в целом.

Особого подхода потребовала унификация БЭВМ, используемых непосредственно для управления различными системами летательных аппаратов. Несмотря на значительный прогресс в области элементной базы, жесткие ограничения на физические характеристики машин этого класса по-прежнему требовали специализации системы команд к особенностям системы управления. Решение этой проблемы было найдено ***благодаря оригинальной архитектуре «Поиск»*** (Проблемно-Ориентируемая с Изменяемой Системой Команд), позволяющей адаптировать набор команд к решаемым задачам, путем расширения основного набора за счет команд, свойственных конкретным задачам мобильных систем [26].

Архитектура «Поиск» включала в себя четыре группы команд: операторы ядра типа обычных команд, операторы более сложной структуры, специальные операторы (обмена, операционной системы) и операторы пользователя. Разрядность операторов переменная. В зависимости от области применения число операторов в системе команд колебалось от 157 до 256. Как показали исследования и опыт эксплуатации, БЭВМ архитектуры «Поиск» при одинаковой элементной базе превосходили обычные одноадресные архитектуры по производительности в 1,5–2,5 раза,

а по компактности кода в 3–5 раз.

На основе унифицированных архитектур в ходе работ второго этапа по созданию ряда «Аргон» (середина 1970-х – конец 80-х годов) ***было предложено несколько моделей машин:*** А-30, А-40, А-50 (архитектура ЕС ЭВМ), Ц100, Ц101, Ц102 (архитектура «Поиск»). Эти машины проектировались в расчете на ***крупносерийное производство и широкое применение в оборонных системах.*** В связи с этим первостепенное внимание уделялось снижению трудоемкости их изготовления и стоимости, обеспечению контроле– и ремонтопригодности, и удобства эксплуатации, созданию моделей межвидового применения, устойчивых к внешним воздействиям применительно к нескольким группам эксплуатации оборонной техники.

БЭВМ А-30— первая модель из ряда унифицированных высокопроизводительных 32-разрядных бортовых ЭВМ архитектуры ЕС ЭВМ, предназначены для обработки и хранения больших массивов информации. Она была спроектирована на основе принятых в ЕС ЭВМ: архитектуры, структурной организации, схемотехнических и конструктивно-технологических решений. А-30 полностью информационно и программно (снизу-вверх) совместимая с ЕС ЭВМ. В машине реализован стандартный набор команд ЕС ЭВМ за исключением команд десятичной арифметики и команд над операндами с плавающей запятой. Машина построена с максимальным использованием принципов модульности и стандартизации блоков, что позволяло гибко изменять вычислительные возможности. Для повышения быстродействия в ней реализовано трехуровневое совмещение операций. Система ввода-вывода включала два мультиплексных канала (специализированный и ЕС ЭВМ) и обеспечивает высокоскоростной обмен информацией с абонентами в реальном времени.

БЭВМ А-40 представляет собой среднюю модель ряда высокопроизводительных 32-разрядных бортовых ЭВМ архитектуры ЕС ЭВМ, являющуюся дальнейшим развитием модели А-30. Основные усовершенствования: полное соответствие архитектурным концепциям ЕС ЭВМ, возможность подключения дополнительных каналов ввода-вывода, а также внешней памяти и устройств ввода-вывода ЕС ЭВМ. В процессоре реализована сложная структура, рассчитанная на совмещение во времени выполнения нескольких команд, ***близкая к структуре ЭВМ ЕС 1060.***

БЭВМ А-50— старшая модель из ряда унифицированных высокопроизводительных 32-разрядных бортовых ЭВМ архитектуры ЕС ЭВМ. Вместе с тем применение более

современной элементной базы позволило резко повысить производительность машины и объем ее оперативной памяти, увеличить число каналов ввода-вывода. В состав машины введен пульт управления с реализацией последовательного интерфейса, процессор содержит кэш-память и микротестовую систему. В оперативной памяти и постоянной памяти микропрограмм реализован контроль с обнаружением двойных и коррекцией одиночных ошибок. Кэш-память имела оригинальную структуру, включающую буфер команд и буфер данных. На базе БЭВМ А-50 была создана ***четырехмашинная вычислительная система*** для ***авиационного комплекса радиолокационного дозора и наведения – А-50.*** В состав комплекса помимо четырех машин с адаптерами канал-канал, объединенных симметричной системой межмашинных связей, входят системный пульт прямого управления и внешний синхронизатор, служащий генератором меток для таймеров всех БЭВМ. Заданная производительность вычислительной системы обеспечивается благодаря распределению задач между отдельными машинами и распараллеливанию алгоритмов.

БЭВМ Ц100, Ц101, Ц102 с архитектурой «Поиск» разрабатывались с конца 70-х годов для удовлетворения потребностей ***отечественной истребительной авиации.*** Их система команд оптимизирована для решения задач управления вооружением на борту истребителей. Выбор соответствующей системы команд (операторов) проводился НИИ «Аргон» совместно с организациями-разработчиками бортовых радиоэлектронных систем. БЭВМ Ц100, Ц101, Ц102 являлись 16-разрядными, синхронными, многоадресными машинами параллельного действия. Эти машины, сочетающие большие вычислительные возможности, компактность конструкции и высокую надежность, – одна из самых удачных разработок в классе авиационных машин. По масштабу производства (выпущено более 4 тыс. экземпляров) принадлежали к числу самых массовых в мировой практике авиационных БЭВМ. Программные средства всех БЭВМ были ориентированы на решение специальных, функциональных задач систем вооружения, и разрабатывались с использованием типовых технологических средств стационарных, универсальных ЕС ЭВМ.

С середины 80-х годов осуществлялись работы ***третьего этапа БЭВМ ряда «Аргон».*** В 1986-м году была принята государственная программа проектирования унифицированных семейств бортовых ЭВМ (СБ ЭВМ) на основе архитектур ЕС ЭВМ, «Поиск» и СМ ЭВМ. В НИИ «Аргон» разрабатывались четыре модели СБ ЭВМ: совместимая с ЕС ЭВМ-2 машина СБ 1180;

одноплатная встраиваемая модель СБ 5580 и четырехпроцессорный вычислительный комплекс СБ 5540 для авиационных и корабельных АСУ (архитектура «Поиск»); модель СБ 3580 для мобильных наземных систем (архитектура СМ ЭВМ). В этих моделях был реализован ряд оригинальных технических решений, но они ***не были запущены в производство по причинам экономической «смуты и развала» в стране.***

Глава 3. История технологических систем для производства программных продуктов в 1960-е – 80-е годы

3.1. История технологической программной системы ЭВМ «Урал»

Технологическое программное обеспечение ЭВМ «Урал» в 1964-м году находилось на достаточно высоком уровне, о чем свидетельствует акт Государственной комиссии, подписанный академиком А.А. Дородницыным: «Впервые в СССР реализован системный подход к разработке математического обеспечения для ряда ЭВМ. В разработанной системе использованы собственные оригинальные решения. Разработанная операционная система выполняет основные функции, реализуемые в современных операционных системах. Документация по математическому обеспечению отличается высоким качеством, полнотой и единством оформления».

Основу системы программного обеспечения семейства «Урал» составляла универсальная программа-диспетчер, выполняющая функции операционной системы. Она обеспечивала ввод и вывод информации, организацию многопрограммной работы, защиту областей оперативной памяти, динамическое распределение оперативной памяти, а также внешней памяти на магнитных барабанах и лентах. С машиной поставлялся автокод АРМУ (Автокод ряда машин Урал), который был единым для ряда ЭВМ типа «Урал». Он был разработан с учетом особенностей этих машин и обеспечивал полную ***совместимость от меньшей машины к большей.***

Каждая ЭВМ «Урал» имела собственной транслятор с языка АРМУ на свой машинный язык. Таким образом, совместимость ЭВМ типа «Урал» была ограниченной и существовала ***только на уровне автокода АРМУ.***

Язык АРМУ обеспечивал: запись программ для работы со словами и массивами переменной длины; выполнение операций над числами в двоичной и десятичной системах счисления; с плавающей и фиксированной запятой. Имелись программы отладки на уровне языков машин и автокода АРМУ, для обнаружения неисправностей ЭВМ был набор тест-программ. Помимо тест-программ, библиотеки стандартных программ и программы отладки с языка АРМУ, с машиной поставлялся транслятор с языка АЛГАМС на АРМУ. Библиотека программ, содержащая стандартные программы и программы решения различных задач, комплектовалась из программ, написанных на языках отдельных ЭВМ, а также на АРМУ, АЛГОЛ-60, АЛГАМС и АЛГЭК. Предусмотрено расширение библиотеки за счет

программ, написанных на других языках и автокодах, после разработки соответствующих трансляторов с этих языков на язык АРМУ.

3.2. История операционной программной системы ЭВМ БЭСМ– 6

В 70-е годы в течение 3 – 5-ти лет, почти одновременно, разрабатывались и были апробированы более пяти крупных операционных систем (ОС) для ЭВМ БЭСМ-6 (рис. 2). Эти ОС были достаточно универсальными, однако несколько различались функциями, языками программирования и ориентировками на специфические особенности применения разрабатываемых, обычно относительно небольших комплексов программ. Усилия концентрировалось на разнообразных языках программирования, на особенностях и эффективности компиляторов, а также на средствах тестирования программных компонентов. Создателей таких ОС, по-видимому, не интересовали в те годы крупные проекты сложных комплексов программ, для которых ***впоследствии оказались необходимы методы и инструментальные средства программной инженерии.*** Поэтому в большинстве ОС отсутствовали средства для системного анализа спецификаций компонентов, планирования и проектирования крупных комплексов программ. Не уделялось внимания созданию методов и средств технико-экономического обоснования проектов прикладных программ, организации коллективов специалистов, контролю реализации, оцениванию и удостоверению качества компонентов и программных продуктов. Также обычно отсутствовали средства обеспечения комплексной отладки и управления конфигурацией сложных комплексов программ. В целом в 70-е годы ***только созревали объективные потребности для создания методов и полноценного инструментария программной инженерии.***

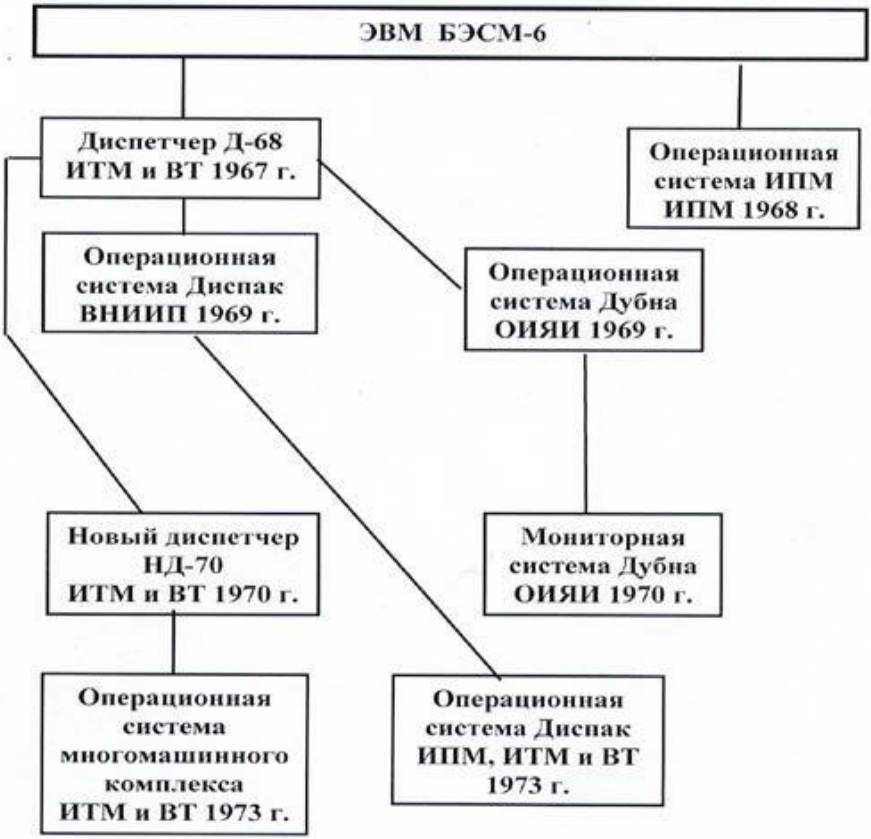

Рис. 2

Первые машины БЭСМ-6 предназначались для установки в центрах, обладавших наиболее сильными коллективами специалистов в области программирования и использования вычислительных машин. Появилась возможность реализовать многие созревшие к тому времени идеи на машине, обладавшей необходимыми аппаратными возможностями для организации мультипрограммирования, режима разделения времени. Коллективными усилиями советских программистов уже к 1968-м году была создана ***система программного обеспечения,*** включавшая в свой состав операционную систему пакетной обработки, трансляторы с машинно-ориентированных языков и с универсальных языков АЛГОЛ-60 и ФОРТРАН. На протяжении всего времени существования машины БЭСМ-6 ее программное обеспечение непрерывно совершенствовалось, и по качеству, объему и возможностям не уступало программному обеспечению лучших отечественных и зарубежных серийных ЭВМ того времени.

Был накоплен огромный фонд программ пользователей и опыт его эксплуатации. На базе БЭСМ-6 был создан ***«золотой фон/»***> программного обеспечения, значение которого трудно переоценить. Появление ко второй половине 60-х годов ЭВМ с аппаратной поддержкой многозадачности и управления параллельной работой устройств, стимулировало создание для этих ЭВМ ***операционных (управляющих) программных систем.*** Первые такие операционные системы (ОС) в СССР, были созданы в организациях, имевших высокий научный и конструкторский потенциал: в Институте точной механики и вычислительной техники, в Институте прикладной математики Академии наук СССР.

В ИТМ и ВТ группу пионеров разработки ОС для БЭСМ-6 [7,11] возглавил Лев Николаевич Королев. Он входит в круг учёных, положивших начало развитию программирования и вычислительной техники в СССР. В 1967-м году он защитил докторскую диссертацию на основе своих работ для вычислительной системы, поддерживающей функционирование системы противоракетной обороны (ПРО). Кроме того, Королёв разрабатывал математическое обеспечение для управления космическими полётами, в том числе для полёта Союз-Аполлон.

В 1967-м году под руководством Л.Н. Королёва была разработана операционная система «Диспетчер-68» для ЭВМ БЭСМ-6, которая стала первой советской мультипрограммной операционной системой. Кроме того, система поддерживала страничную организацию памяти с динамическим распределением памяти, параллельную работу устройств ввода/вывода. Эти исследования значительно повлияли на дальнейшее развитие отечественного системного программирования. Основными участниками разработки Д-68 в ИТМ и ВТ были: Виктор Петрович Иванников и Александр Николаевич Томилин. В ИПМ в этих работах участвовали Михаил Романович Шура-Бура и основные разработчики И.Б. Задыхайло, Э.З. Любимский и В.С. Штаркман. Существенный вклад в развитие системных программных средств был внесен также группой специалистов из лаборатории вычислительной техники и автоматизации Объединенного института ядерных исследований (ОИЯИ) в Дубне под руководством Николая Николаевича Говоруна. Эти ОС обеспечивали параллельное выполнение процессов обработки информации и их иерархическую организацию, интерактивный режим работы коллектива пользователей ЭВМ и обработку информации в режиме реального времени.

Большое влияние на развитие ОС для ЭВМ оказало создание в 1967-м году в ИТМ и ВТ под руководством Л.Н. Королева [7, 11]

первой ОС для ЭВМ БЭСМ-6, названной позднее ***«Диспетчер-68».*** Она обеспечивала совместное управление работой устройств ЭВМ, подготовку и решение задач в мультипрограммном режиме. По современным понятиям ***Д-68*** являлся ядром операционной системы, обеспечивавшим: мультипрограммный режим пакетной обработки заданий, управление виртуальной памятью, управление внешними запоминающими устройствами и многочисленными устройствами ввода-вывода. Государственная комиссия принимала машину БЭСМ-6 в комплексе с ее программным обеспечением, что явилось новым прецедентом приемки вычислительной техники.

В разработке программного обеспечения ЭВМ наиболее трудоемкой и ответственной частью являлось создание ядра операционной системы. В то время в нашей стране не было опыта разработки ОС, а зарубежным опытом нельзя было воспользоваться, поскольку БЭСМ-6 не являлась копией ни одной машины, что определяло значимость успехов в создании отечественных ОС. Было доказано, что сложнейшие ОС ***могут разрабатывать отечественные программисты. Операционная система выполняла:***

мультипрограммное решение задач;

• управление одновременной работой всех каналов связи с внешними запоминающими устройствами и всех устройств ввода-вывода информации;

• совмещение вычислений во всех задачах с параллельной работой внешних запоминающих устройств и устройств ввода-вывода;

• организацию совместного динамического распределения ресурсов двухуровневой памяти оперативной и внешней), базирующуюся на замещении страниц в оперативной памяти;

• распределение устройств между задачами;

• буферизацию ввода-вывода;

• развитую связь с оператором по управлению прохождением задач и работой устройств;

• возможность многотерминальной работы в диалоговом режиме.

Кроме этих основных функций «Диспетчер-68» обеспечивал ***вызов трансляторов*** с языков программирования и автокодов. Д-68 явился предтечей будущих развитых ОС и основой последующих операционных систем для БЭСМ-6: ОС «Дубна» (Н.Н. Говорун) и ОС «Диспак» (В.Ф. Тюрин), дисковой операционной системы, ориентированной на пакетную обработку и эксплуатируемой в дальнейшем на большинстве ЭВМ БЭСМ-6.

После появления у пользователей нескольких ведущих

институтов машины БЭСМ-6 и ОС Д-68, их программистские коллективы начали ***активную разработку собственных ОС,*** с их точки зрения более адекватно отражавших использование вычислительной техники в этих институтах. Дублирование работ по созданию различных ОС для машины одного типа было полезным, прежде всего тем, что на этих работах выросли специалисты системного программирования высочайшей квалификации и оформились известные в нашей стране программистские школы: ИПМ им. М.В. Келдыша, ИТМ и ВТ им. С.А. Лебедева, Новосибирская школа, возглавлявшаяся А.П. Ершовым, школа ОИЯИ (Дубна), которой руководил Н.Н. Говорун, школа в НИВЦ МГУ, а также ряд других школ, включая ***ядерные и оборонные центры страны.*** В результате в 70-е годы одновременно в эксплуатации находились: ОС Д-68, ОС ИПМ, ОС ДИСПАК, ОС ДУБНА, НД-70.

ОС Д-68, как и другие ОС БЭСМ-6, в ходе эксплуатации расширял свои возможности в части услуг, предоставляемых пользователям. Такие системы как ПУЛЬТ, МУЛЬТИДОСТУП, ДИМОН (диалоговый монитор) обеспечивали режим многопультового доступа к машине с удаленных терминалов. Появились также системы отладки программ в терминах входного языка, обеспечивающие связь задачи пользователя с терминалами; система для редактирования файлов и запуска задач в решение с удаленных пультов. К началу 70-х годов в состав программного обеспечения БЭСМ-6 входили все основные универсальные языки программирования: АЛГОЛ-60, ФОРТРАН, ЛИСП. Первый транслятор с языка Фортран для БЭСМ-6 был разработан в 1969-м году в ОИЯИ в Дубне. Этот транслятор затем был включен в Мониторную систему «Дубна».

В различных организациях использовалось несколько вариантов трансляторов с языков, генерирующих программы разной степени эффективности, в том числе оптимизирующие трансляторы с языков АЛГОЛ-60 и ФОРТРАН и компилятор с языка ЛИСП. В состав программных средств машины БЭСМ-6 входил спектр проблемно-ориентированных языков СИМУЛА-67, ГРАФОР, ГРАФАЛ, язык типа EPSILON и ряд других специализированных языков. Для построения диалоговых систем использовались пошаговые трансляторы с некоторых других языков. Для разработки и отладки комплексов программ мобильных и бортовых ЭВМ поставлялись кросс-системы ЯУЗА-6 и ТЕМП (см. п. 3.4). По критериям своего времени ***программное обеспечение БЭСМ-6 было наиболее развитым*** в сравнении с обеспечением других машин отечественного производства и по принципиальным возможностям не уступало программному

обеспечению многих зарубежных машин.

Существенное влияние оказал «Диспетчер-68» на разработку в ИТМ и ВТ операционной системы реального времени для БЭСМ-6 ОС ***НД-70 («Новый диспетчер-70»)*** с развитыми средствами организации параллельных вычислений (соподчинение задач, аппарат параллельных процессов), режимом работы в реальном времени и возможностью организации многомашинного вычислительного комплекса (руководитель В.П. Иванников). В центрах управления полетами космических аппаратов были созданы и в течение более двадцати лет активно использовались ОС НД-70 для обеспечения управления полетами, а также для нескольких больших баллистических и телеметрических программных комплексов реального времени. Вслед за НД-70 средства организации параллельных процессов были введены в ОС «Диспак», что позволило программным комплексам реального времени базироваться на этой ОС. На основе возможностей операционной системы БЭСМ-6 усилиями ряда ведущих научноисследовательских и производственных организаций с привлечением крупных математических сил СССР была создана, первая в стране ***поставляемая промышленностью полная система математического обеспечения ЭВМ*** (ОС, системы программирования, библиотеки программ).

В 1967-м году И.Б. Задыхайло, С.С. Камынин и Э.З. Любимский разработали [2, 7] ***операционную систему ИПМ*** АН СССР (ОС ИПМ) под руководством Э.З. Любимского. При разработке ОС ИПМ авторы широко использовали механизмы для организации взаимосвязи между задачами и процессами. Каждый ресурс (память, файл, устройство) имел своего хозяина, который мог его отдавать или сдавать в аренду любой другой задаче, оговаривая соответствующие права использования, в том числе и право дальнейшей передачи в аренду. ОС ИПМ органично включала в себя систему программирования, что позволило довольно легко обеспечить такие свойства, как шаговая трансляция и отладка в терминах языка. Большинство трансляторов были написаны на языке АЛМО (аналог языка Си), и использовали его в качестве выходного языка. Это позволило сначала раскрутить и отладить их на машине М-220, а затем (в 1969-м году) перенести на БЭСМ-6 в среду ОС ИПМ, что избавило разработчиков трансляторов и операционной среды от многих излишних взаимных претензий.

Многоязыковая ***Мониторная система «Дубна»*** для БЭСМ-6 была разработана Н.Н. Говоруном, В.П. Шириковым в ОИЯИ в Дубне [7]. (1970-й год), обеспечивала управление заданиями, создание и использование многоуровневых библиотек программ.

В систему входила библиотека программ общего назначения, совместимая с библиотекой Европейского центра ядерных исследований CERN. Мониторная система «Дубна» использовалась и с другими ОС для БЭСМ-6. ***В Мониторную систему «Дубна» входили следующие компоненты***:

• транслятор (ассемблер) с автокода Мадлен на язык загрузки;

• транслятор с языка Фортран на язык загрузки;

• статический и динамический загрузчики;

• библиотекарь и общие библиотеки стандартных программ;

• редактор текстовой информации;

• системные программы ввода-вывода.

• В дальнейшем в состав Мониторной системы ***были включены трансляторы и системы***:

• Алгол-ГДР;

• Фортран-ГДР;

• Форекс оптимизирующий транслятор с языка, близкого к Фортрану 77;

• транслятор с языка Паскаль;

• Графор пакет графических программ;

• Поплан транслятор с языка РОР-2.

Мониторная система «Дубна» была создана коллективом сотрудников ОИЯИ с участием специалистов из Института атомной энергии им. И.В. Курчатова и стран-участниц ОИЯИ (ГДР, ВНР, КНДР). В дальнейшем развитии системы приняли участие также сотрудники ИК АН УССР, ИАПУ ДВНЦ АН СССР, ИФВЭ и других организаций.

В 70-х годах под руководством Л.Н. Королева и В.П. Иванникова впервые была создана ***распределенная ОС многомашинного комплекса,*** обеспечивающая сетевое взаимодействие вычислительных процессов в ЭВМ комплекса, а также с процессами в глобальных сетях ЭВМ, и использование внешних устройств ЭВМ в любых вычислительных процессах, выполняющихся в комплексе. Была фактически обеспечена работа «конвейера ЭВМ», предназначенного для обработки ***в режиме реального времени больших потоков информации о полетах космических аппаратов.***

Все эти разработки для ЭВМ БЭСМ-6, которая более десяти лет оставалась самой высокопроизводительной машиной в стране, и для многомашинного вычислительного комплекса реального времени АС-6, обеспечили обработку информации в центрах управления космическими полетами, во многом определили дальнейшие направления и характер исследований в отечественном системном программировании. За время

эксплуатации нескольких сотен БЭСМ-6 была накоплена ***уникальная библиотека программ, которая стала беспрецедентным интеллектуальным богатством страны.***

Ведущими ***разработчиками программного обеспечения АС-6*** были В.П. Иванников и А. Н. Томилин. Система функционировала в режиме дистанционной пакетной обработки, в режиме коллективного пользования и в режиме реального времени. ***АС-6 обладала трехуровневой структурой.*** Первый уровень составляют высокопроизводительные процессоры ЭВМ, блоки оперативной памяти и средства для объединения этих устройств в единый комплекс. Устройства, входящие в эту группу, связываются между собой при помощи наиболее быстрого канала (канал 1-го уровня). В состав высокопроизводительных устройств входили новые центральные процессоры – ЦП АС-6 и ЭВМ БЭСМ-6.

Подготовка информации для центрального комплекса является основной задачей второго уровня АС-6, который представляет собой аппаратно-программный комплекс, обеспечивающий подключение к центральному комплексу внешних накопителей, устройств ввода-вывода, каналов связи и средств отображения. Средством связи внешних устройств с оперативной памятью служил унифицированный канал 2-го уровня.

Типовое программное обеспечение системы АС-6 состояло из следующих основных компонентов:

• операционной системы (ОС) АС-6, объединявшей ОС центральных процессоров АС-6, ОС БЭСМ-6 и ОС периферийных машин (ПМ-6) АС-6;

• операционная система АС-6 была реализована таким образом, что допускала взаимодействие до 16 процессоров и ЭВМ, подключенных к каналу 1-го уровня;

• систем автоматизации программирования;

• тестов и обслуживающих программ.

В дальнейшем была создана совместимая с БЭСМ-6 новая машина – ***«Эльбрус Б»*** («интегральная БЭСМ-6»), на порядок более быстрая, чем БЭСМ-6 [2, 11]. Машинное слово ее могло быть 48-разрядным, как на БЭСМ-6, так и 64-разрядным, как у большинства суперЭВМ. В этом случае за счет более длинного адреса существенно увеличивалось адресное пространство виртуальной оперативной памяти. В эскизном проекте Эльбрус (1970-й год) было показано, что основной путь дальнейшего повышения производительности вычислительных систем лежит в распараллеливании процесса вычислений. В этой связи было решено разработать модульный масштабируемый

вычислительный комплекс, комплектацию которого заказчик определял в зависимости от специфики использования. Однако модульная архитектура многопроцессорного вычислительного комплекса (МВК) использовалась не только для повышения общей производительности, но и для повышения надежности вычислений.

3.3. История разработки основной операционной системы ОС ЕС

В ***августе*** 1968-го года в НИЦЭВТе было сформировано отделение системного (технологического) программного обеспечения [24]. В 1970-м году его возглавил Леонид Дмитриевич Райков, руководивший работами по программному обеспечению семейства ЕС ЭВМ бессменно более 20 лет.

В ГДР работали несколько машин IBM-360, и специалисты комбината «Роботрон» имели хороший опыт освоения ДОС-360 и ОС-360. В ***начале*** 1970-го года комбинат «Роботрон» принял делегацию специалистов НИЦЭВТ, которая сгенерировала на машине IBM-360 технологический вариант операционной системы. Результатом последующих командировок программистов НИЦЭВТ в ГДР были ***распечатки молу лей ОС IBM-360 на языке Ассемблер,*** полученные с помощью специально разработанной программы. В том же 1970 м году был оформлен контракт на совместную разработку ДОС ЕС и ОС ЕС специалистами НИЦЭВТ и комбината «Роботрон». Был найден удачный вариант контракта, когда коллективы, работая отдельно на своих площадях и при собственном финансировании, производили половину общего объема работ и результаты передавали партнеру. Таким образом, при выполнении своих обязательств каждая сторона получала полный продукт при половинном финансировании.

В 1970-м году в ВЦ НИЦЭВТ появились две модели IBM-360/40, а в сентябре 1972-го года к ним добавилась модель 145 семейства IBM-370. Машины были куплены в ФРГ ***в обход существовавшего эмбарго*** западных стран и существенно ускорили процесс создания ОС ЕС, а главное – обеспечили контроль её совместимости с ОС-360. В ***апреле*** 1973-го года ***издание 10 ОС ЕС*** прошло государственные испытания вместе с ЭВМ ЕС-1050. Она предоставляла пользователю два режима пакетной обработки данных – однопрограммный и мультипрограммный с фиксированным (до 15) числом задач. Система была модульной и открытой для расширения. В ее состав помимо комплекса программ, управляющих данными, заданиями и задачами, входили компиляторы с языков Ассемблер, РПГ, Фортран, Алгол-60, ПЛ-1, Кобол. Общий объем системы составил ***2 млн. команд.*** В 1975-м году появилось издание 4.0, предоставлявшее пользователю три режима пакетной обработки данных: однопрограммный, многопрограммный с фиксированным

числом одновременно решаемых задач (MFT) и новый – многопрограммный с переменным числом одновременно решаемых задач (MVT).

Важным этапом в развитии ОС ЕС был выпуск в 1976-м году ***издания 4.1,*** одного из наиболее популярных среди пользователей ЕС ЭВМ. В ОС ЕС издания

4.1 ***впервые появились:***

диалоговый удаленный ввод заданий;

• общий телекоммуникационный метод доступа;

• обеспечение дисплеев в качестве консоли оператора;

• обеспечение 29 Мб накопителей на магнитных дисках;

• программное обеспечение графопостроителей.

ОС ЕС издания 4.1 насчитывала около ***3 млн. команд,*** в ее состав входили 90 документов общим объемом более ***16 тыс. страниц*** [24].

В 1978-м году была закончена разработка ***ОС ЕС 6.0*** для второй очереди машин ЕС ЭВМ.

Система предоставляла пользователям ЕС ЭВМ три режима пакетной обработки: мультипрограммирование с фиксированным и переменным числом задач и новый режим виртуальной памяти, обеспечивающий 16 Мб виртуальной памяти для самой системы и пользователей ЭВМ. В ***ОС ЕС 6.0 впервые появились:***

программное обеспечение 100 Мб накопителей на магнитных дисках;

• обеспечение режима разделения времени;

• обеспечение средств комплексирования моделей ЕС ЭВМ (адаптера канал-канал, средств прямого управления и общего поля внешней памяти);

• монитор динамической отладки;

• универсальное средство трассировки.

В ОС ЕС издания 6.0 получили дальнейшее развитие диалоговый удаленный ввод заданий (обеспечение дисплея ЕС 7906), средства восстановления для моделей ЕС 1060 и ЕС 1035, средства машинной графики и телеобработки (графический и базисный телекоммуникационный методы доступа в режиме виртуальной памяти), измерения и учета (системная мониторная программа в режиме виртуальной памяти).

С 1978-го года разрабатывалась ***ОС ЕС издания 6.1 —*** дальнейшее развитие системы 6.0. В этом издании ОС ЕС были обеспечены все модели ЕС ЭВМ-2 в полном объеме, развиты все компоненты ОС для режима виртуальной памяти, в том числе: средства комплекси-рования, диалоговый удаленный ввод заданий, общий телекоммуникационный метод доступа, обеспечение режима разделения времени. По мере развития ОС

6.1 разработаны девять ее модификаций, в которых усовершенствованы алгоритмы управляющей программы, расширена номенклатура периферийных устройств. В ОС 6.1 включено обеспечение многомашинной подсистемы ввода заданий РОС. Для нее были разработаны новые, усовершенствованные системы программирования Фортран, Кобол, ПЛ-1 (отладочный и оптимизирующий), подсистема ввода заданий КРОС, расширенная Сортировка-Объединение. ***ОС ЕС 6.1 была, самой распространенной и популярной из основных операционных систем ЕС ЭВМ.***

ОС 7 ЕС издания 01 появилась в 1983-м году. В ней были усовершенствованы принципы функционирования предшествующих изданий ОС ЕС путем введения в систему в качестве основы ***концепции виртуальной машины.*** Это позволило упростить внутреннюю структуру ОС и благодаря этому повысить эффективность ее функционирования при сохранении программной совместимости на пользовательском уровне с предыдущими изданиями ОС ЕС. В состав системы ОС 7 ЕС входили две составные части: система виртуальных машин (СВМ ЕС разработки НИИЭВМ) и базовая операционная система (БОС). ***СВМ*** обеспечивала основу для функционирования ОС-7 ЕС путем реализации концепции виртуальных машин. С помощью виртуальных машин осуществлялись все режимы и подсистемы ОС-7 ЕС. СВМ ЕС с помощью подсистемы диалоговой обработки (ПДО) реализовала режим диалога.

БОС была предназначена для создания ОС и выполнения основных функций ОС ЕС. Она ориентирована на режим пакетной обработки и могла быть использована только в среде виртуальных машин. При этом сохранялась преемственность и программная совместимость с ОС ЕС издания 6.1 по программам пользователя. Ориентация БОС на функционирование только в среде виртуальных машин СВМ ЕС, определяла ряд ее особенностей по отношению к режимам управляющей программы (MFT, MVT, SVS) ОС ЕС издания 6.1. В ***состав БОС входили программы***:

- управления заданиями, работой, задачами, данными;
- обеспечения машинной графики и средств телеобработки;
- системные обрабатывающие программы;
- средств восстановления, измерения и учета, генерации.

Для работы в сетях ЭВМ было разработано программное средство ***ПРИСС,*** представляющее собой ***систему прикладных сетевых служб.*** Система предоставляла эффективные средства организации обработки и передачи файлов и доступ широкому кругу удаленных пользователей к виртуальным

файлохранилищам и к службам виртуальных заданий в сети ЭВМ.

Система позволяла:

выполнять обработку и передачу файлов между абонентами сети, используя команды абонента и макрокоманды прикладных программ;

• организовать передачу заданий, распределение работ по выполнению заданий в сети и выполнение заданий;

• разрабатывать прикладные программы для использования служб виртуального файла и виртуального задания.

До 1989-го года было выпущено ***четыре издания ОС 7 ЕС***\ соответственно четырем изданиям системы СВМ. Подобная ***структура ОС 7 полностью оригинальна и не имела аналогов за рубежом*** [24].

Мультивиртуальная операционная система (МВС ЕС) разрабатывалась в НИЦЭВТ с 1984-го года в кооперации с комбинатом «РОБОТРОН» коллективом программистов под непосредственным руководством В.Г. Лесюка. Для ЕС ЭВМ она являлась дальнейшим развитием ОС 6.1 ЕС. МВС была предназначена для эксплуатации на старших моделях ЕС ЭВМ-2 и ЕС ЭВМ-3. Каждому пользователю она предоставляет адресное пространство размером 16 Мб вне зависимости от режима работы – пакетная обработка или в режиме разделения времени. В первом случае выполнение задания контролировала подсистема обработки заданий, во втором – монитор абонентского пункта. До 1989-го года были выпущены четыре издания системы МВС ЕС.

Операционная система реального времени (ОС РВ) была разработана в ***мае*** 1985-го года ***(первая версия)*** для бортовых ЭВМ Аргон, создаваемых в НИЦЭВТ и совместимых с ЕС ЭВМ (см. главу 2). Она была построена на базе ППП «Супервизор реального времени» (СРР). ОС РВ представляла собой универсальную операционную систему для управления в реальном масштабе времени устройствами ввода-вывода, в том числе и нестандартными на уровне процессов. Она могла функционировать как в монопольном режиме, так и совместно с ОС ЕС. В монопольном режиме выполнение программ происходит в среде ОС РВ без обращения к средствам ОС ЕС. В этом случае резидентные накопители на магнитных дисках не применялись. Монопольный режим был предназначен для управления вычислительным процессом как на одной ЭВМ, так и на вычислительном комплексе, использующем в качестве средств комплексирования адаптеры канал-канал.

Мобильная операционная система для ЕС ЭВМ (МОС ЕС) – это интерактивная ОС общего назначения с разделением времени. Она представляла средства для создания мобильных

программных комплексов, имеющих единый пользовательский интерфейс и функционирующих на ЭВМ различных архитектур под управлением операционных систем, концептуально совместимых с ОС Unix. Основным языком программирования в МОС ЕС являлся язык Си. Помимо этого, поддерживались языки Паскаль и Фортран-77. Диалог пользователей с МОС ЕС обеспечивал интерпретатор командного языка, сочетающего интерактивные средства с возможностями языка программирования. В МОС ЕС имелся диалоговый отладчик программ, средства программирования на языке Си, средства обмена с дисплейными комплексами ЕС 7920, генераторы синтаксических и лексических анализаторов, универсальный калькулятор, развитые средства редактирования и форматирования текстов, комплекс программ обеспечения межмашинного обмена, а также полностью автоматизированные средства информационного справочного обслуживания.

В течение 1975-го – 80-го годов был разработан ***ряд пакетов прикладных программ*** для применения в составе ОС ЕС. Назначение пакета ***супервизор реального времени (СРВ)*** – обеспечивать эффективное использование моделей ЕС ЭВМ с ОС ЕС, поддерживающих режимы реального времени с жесткими ограничениями на время отклика. Пакет прикладных программ ***«ОКА»*** был разработан в 1980-м году и представлял собой универсальную систему управления базами данных. Пакет предназначен в качестве базового для областей применения, использующих централизованные банки данных в режиме пакетной обработки и в режиме оперативного доступа с удаленных терминалов. Пакет функционировал под управлением ОС ЕС изданий 4.0, 4.1, а также 6.1 в режиме SVS. В 1984-м году закончена разработка ППП «ОКА-ВС», обеспечивающего расширенный набор функциональных возможностей и более эффективное использование виртуальной памяти.

В 1980-м году был закончен третий этап разработки ППП ***«КАМА>,*** в результате в него вошли локальный и удаленный комплексы терминалов ЕС 7920 со средствами форматирования сообщений и было обеспечено функционирование ППП «КАМА» под управлением ОС 6.1 в режиме SVS. Пакет прикладных программ для построения информационно-справочных систем представлял собой универсальный монитор телеобработки данных с развитыми средствами управления данными. Пакет был предназначен для использования при создании систем оперативного сбора и ввода данных, систем удаленной пакетной обработки данных, различного рода информационно-поисковых и информационно-справочных систем, работающих в режиме

одновременного обслуживания широкой сети терминалов, обеспечивающих коллективный доступ и обновление общих массивов информации, интерактивных систем обучения.

Развитие ***операционных систем ЕС ЭВМ было ориентировано*** на возможность создания крупных комплексов программ большими коллективами специалистов предприятий, для автоматизации административных, финансовых, организационных, штабных и иных функций. Это способствовало освоению и массовому переходу в стране специалистов от ***программирования «в малом» к программированию «в большом» крупных программных продуктов.*** Стоимость прикладного математического обеспечения, имеющегося у пользователей для разных типов машин к этому времени должна, была составить уже 8 – 10 млрд. рублей. Широкий набор средств программирования и унификация программного инструментария в операционных системах ЕС ЭВМ позволяли в значительной степени автоматизировано переносить имеющийся задел прикладных программ на ЕС ЭВМ. Таким образом, мог быстро расширяться накопленный состав готовых прикладных программных продуктов [24].

Однако некоторые важнейшие ***концептуальные положения и функции программной инженерии не вошли*** в состав методов и средств операционных систем ЕС ЭВМ, даже последних версий. В них практически полностью отсутствуют средства автоматизации менеджмента и управления качеством крупных проектов программных средств, поддержка формирования и верификации спецификации требований к программным компонентам. Не уделено внимания методологии и поддержке координации и взаимодействия специалистов, совместно работающих над сложными комплексами программ, их сопровождению в процессе развития и модернизации с контролем и гарантированием качества. Тем не менее, ***стратегическое решение*** о развитии ЕС ЭВМ на основе IBM-360, ***позволило сконцентрировать ресурсы*** страны на ***промышленном создании*** необходимой номенклатуры аппаратуры вычислительной техники и инструментальных операционных систем для массовой эффективной разработки программных продуктов и систем автоматизации в различных отраслях народного хозяйства.

3.4. История технологической программной системы ЭВМ М-10 и М-13

Технологическое программное обеспечение машины М-10 включало (1970-е годы): операционную систему, обеспечивающую разделение времени и оборудования, диалоговый режим одновременной отладки до 8 независимых программ и мультипрограммный режим автоматического прохождения до 8 независимых задач. Система автоматизации программирования, включала машинно-ориентированный язык – Автокод и проблемно-ориентированный язык – Алгол-60, соответствующие трансляторы и средства отладки; библиотеку типовых и стандартных программ; диагностические программы; программы контроля функционирования (тесты).

Сразу после установки ЭВМ, до завершения испытаний, проводилась отладка новых боевых программ [9, 11]. Долгое время не удавалось добиться устойчивой работы вычислительного комплекса с новой боевой программой. Каждые два – четыре часа происходили сбои ЭВМ с обнулением информации. Наработка на отказ составляла 10–15 часов. ***Причины сбоев*** разработчики боевой программы объясняли неустойчивой работой ЭВМ, а разработчики ЭВМ – ошибками в боевой программе. Несмотря на все усилия разработчиков, устойчивость функционирования ЭВМ повышалась медленно. Хотя наработка на отказ увеличилась до 40–50 часов, она, все же была значительно меньше той, что задавалась техническими требованиями. При грамотной эксплуатации и отработанной боевой программе количество отказов было впоследствии снижено в несколько раз по сравнению с данными, полученными на испытаниях.

На ***стадии программирования начались дискуссии*** между сторонниками единой, монолитной программы и ее модульного построения (в ***начале*** 70-х годов ***это было удивительно!?)*** [9]. Сначала победили представители первой школы. Но возникли большие трудности отладки при неустойчивой работе ЭВМ. Поэтому программу все же разбили на модули с простыми принципами передачи управления от модуля к модулю. Позже, при вводе КП СПРН, специалисты НИИВК сравнительно быстро повысили надежность ЭВМ М-10 до приемлемого уровня. На стадии комплексной отладки между алгоритмистами и программистами, работающими в разных подразделениях, возникали трения. Один из разработчиков, войдя в роль

неформального лидера, взял на себя управление отладкой, и доработкой алгоритма в процессе отладки.

Еще одна трудность заключалась в том, что отлаживать программу до ввода ЭВМ на объекте было негде – изготавливаемые ЭВМ, шли с завода прямо на объекты. Поэтому монтаж аппаратуры и отладка программ велись последовательно, хотя выполнялись разными коллективами, которые могли бы работать параллельно. К тому же, качество изготовления и степень отладки аппаратуры были низкими, фактически доводка ЭВМ происходила одновременно с отладкой программ. ***Время разработки программ составляло от трети до половины времени создания всего объекта.***

3.5. История технологии программирования для специализированных ЭВМ реального времени на БЭСМ-6 в 1970-е годы

В 70-е годы технологическая ***программная инженерия оказалась наиболее востребованной для решения крупных задач в оборонной технике и для государственных административных систем.*** Массовое прикладное программирование в академических организациях, промышленных институтах и в вузах оставалось на уровне ***индивидуального создания*** небольших программ и обычно не требовало применения мощных методов программной инженерии. Опыт предприятий оборонной промышленности, накопленный в 60-е годы, позволил обобщить особенности и проблемы развития программной инженерии для эффективного создания наиболее сложных ***комплексов программ военного применения, базирующихся на специализированных ЭВМ.***

Быстрый рост сложности функциональных задач и потребных ресурсов ЭВМ для их решения в 60-е годы, не могли быть удовлетворены при доступной в то время технической и элементной базе вычислительных машин. Эту проблему разработчики систем стремились решать путем адаптации архитектуры ЭВМ к особенностям и характеристикам сложных функциональных задач (см. п. 1.3). Наиболее сильно эта тенденция проявилась при необходимости применять вычислительную технику в мобильных системах оборонного назначения. Поэтому во многих организациях оборонной промышленности еще в конце 50-х годов начали разрабатываться многочисленные специализированные ЭВМ. При создании требований к таким объектным ЭВМ военного назначения для эффективного использования их ограниченных вычислительных ресурсов, необходим был детальный анализ алгоритмов и программ, предназначенных для функционирования в реальном времени.

В конце 60-х годов в МНИИПА (Московский НИИ приборной автоматики – НИИ-5 – директор Анатолий Леонидович Лившиц, а затем Владимир Алексеевич Шабалин) был выделен небольшой коллектив для разработки системы автоматизации программирования. Принципиально возможности открылись в начале 70-х годов в связи с появлением в институте ЭВМ БЭСМ-6.

Началось создание ***системы автоматизации разработки и отладки программного обеспечения (САРПО)*** ЯУЗА-6 (под руководством Владимира Васильевича Липаева и Льва Александровича Серебровского). Эти работы были активно поддержаны со стороны руководства института и министерства. Министерством радиопромышленности СССР было открыто достаточное финансирование работ и последовательно значительно увеличен коллектив специалистов. Были сформулированы следующие ***основные концептуальные особенности САРПО ЯУЗА-6:***

возможность автоматизированной настройки системы на структуру команд различных мобильных и бортовых, специализированных ЭВМ;

• три входных языка программирования для специализированных ЭВМ – автокод, макроязык, алгоритмический язык;

• высокое качество программ, транслированных с алгоритмического языка (коэффициент эффективности – расширения кодов программ в пределах 1,1 – 1,2);

• автоматизированная стыковка программных компонентов по глобальным переменным и по передачам управления;

• автоматизированный контроль структурного построения и использования памяти программами;

• автоматизированная отладка компонентов на универсальной технологической ЭВМ на уровне входного языка программирования с интерпретацией команд специализированной ЭВМ;

• автоматический выпуск документации на весь комплекс программ и на отдельные подпрограммы в соответствии с ГОСТами и пригодной для ввода программ в специализированную ЭВМ.

Перечисленные особенности отличали САРПО ЯУЗА-6 от обычных трансляторов как ***комплексную систему программной инженерии,*** предназначенную для автоматизации основных этапов разработки крупных комплексов программ, при условии высокой эффективности создаваемых программ по использованию ресурсов ЭВМ (рис. 3). Комплексная автоматизация должна была в несколько раз повысить производительность труда при разработке больших управляющих программ (размером порядка 100 тыс. команд). При этом должны были исследованы, и учитываться принципиальные особенности мобильных, специализированных ЭВМ и комплексов программ оборонных систем того времени [16, 18].

Значительное расширение функций автоматизации

разработки программ на мощных технологических ЭВМ (БЭСМ-6), а также ***быстрый рост числа различных структур мобильных и бортовых ЭВМ,*** для которых была необходима автоматизация программирования, приводили к сокращению общей доли, изменяемой части инструментальных систем разработки программ для множества применяемых ЭВМ. В результате было показано, что ***экономически целесообразно*** выделять в инструментальных технологических системах программирования ***машинно ориентированную на специализированные ЭВМ часть программ,*** и отдельно автоматизировать их разработку. Впоследствии, примером могла служить система ЯУЗА-6, в которой из 400 тысяч команд, для настройки на различные специализированные ЭВМ оказалось необходимо всего около ***3–5 % изменяемых команд-*** Для автоматизации такой настройки целесообразно было разработать дополнительную подсистему размером около 10–20 тысяч команд и соответствующие инструкции по ее применению.

Основное достоинство разработанного и исследованного оригинального метода адаптируемых кросс-систем состояло в том, что эти системы могли быть ориентированы практически на любые типы специализированных ЭВМ, поддерживая при этом эффективность результатов программирования на достаточном уровне. Процесс адаптации кросс-систем был формализован и автоматизирован.

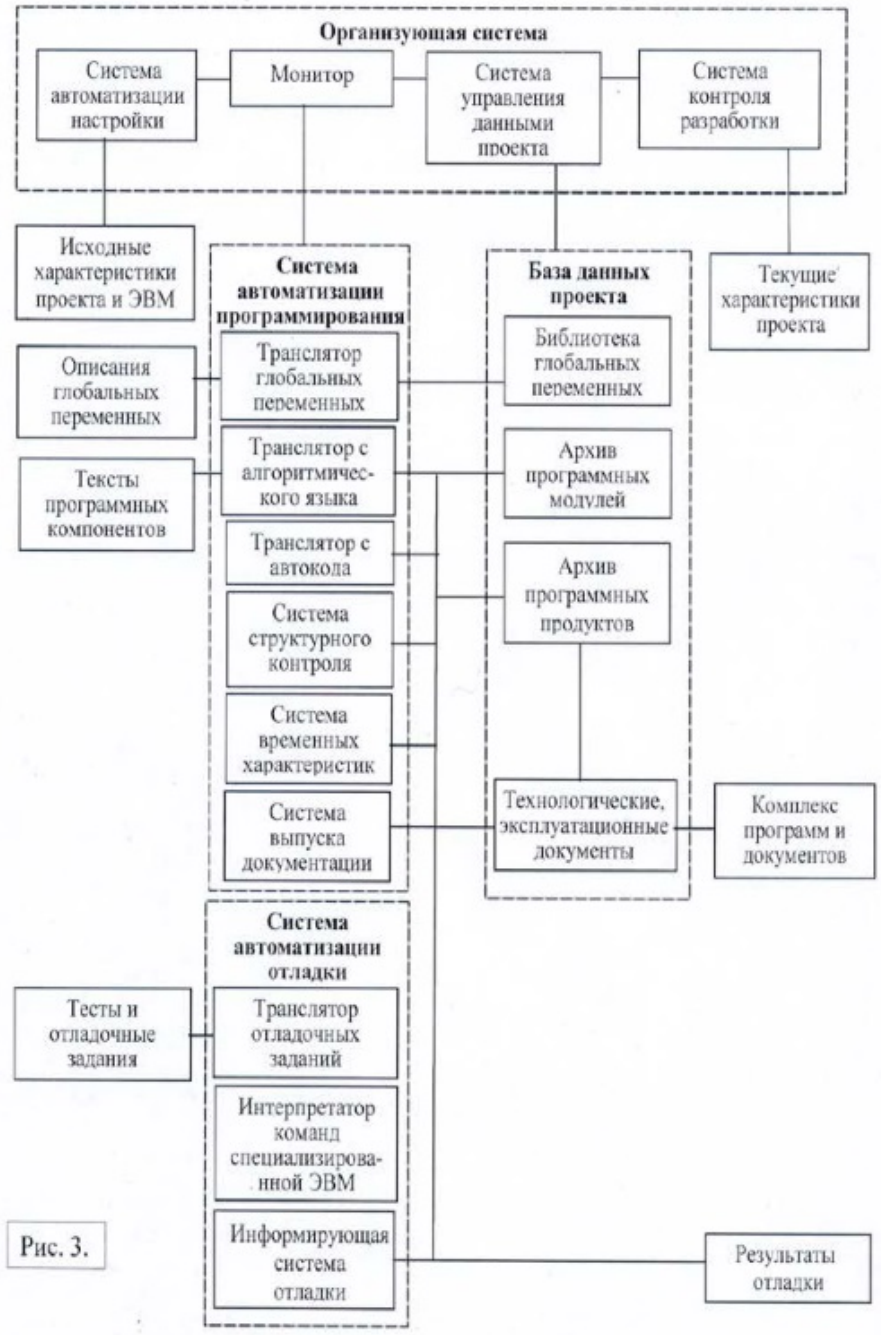

Рис. 3

При разработке адаптируемой кросс-системы в ее машинно-зависимые технологические программы были введены проблемно-ориентированные средства, предназначенные для обеспечения

автоматизации адаптации.

Для формирования подсистемы адаптации было ***исследовано множество характеристик и параметров, различных специализированных ЭВМ.*** При этом был использован не только статистический материал, накопленный при рассмотрении специализированных ЭВМ, но и априорные ***обобщенные характеристики широкого класса объектных вычислительных машин.*** В пределах этого класса определялись типы данных, с которыми работают специализированные ЭВМ, специфика систем команд, их форматы и времена исполнения; состав памяти данных и команд; системы адресации и т. д. Данные характеризовались типом, формой представления, способом кодирования и точностью.

При полной трудоемкости создания ЯУЗА-6 около 300 человеко-лет, для адаптации всей системы на архитектуру любой специализированной ЭВМ, требовалось только около одного человеко-года квалифицированного специалиста.

К программным средствам объектных ЭВМ реального времени в 70-е годы предъявлялись особенно жесткие требования к использованию их вычислительных ресурсов, что вызывало необходимость построения и применения специфических методов программирования и соответствующих языков для систем реального времени. Для сложных комплексов программ было характерно разнообразие алгоритмов решения функциональных и вспомогательных задач. При этих условиях трудно было удовлетворить требования высокой эффективности получаемых программ и одновременно высокой производительности труда специалистов, если проводить программирование на одном языке. Проведенные ***исследования показали, что перспективным является путь автоматизации программирования на базе группы взаимно связанных входных языков,*** различающихся глубиной машинной и проблемной ориентации, применяемых в зависимости от типов задач. По своей ориентации языки программирования ***полезно было разделить*** на:

• ***автокоды,*** предназначенные для записи программ с полным учетом структуры и системы команд конкретной специализированной ЭВМ;

• ***макроязыки,*** предназначенные для записи программ с использованием унифицированных операторов и процедур, состав и содержание которых учитывают не полностью структуру каждой специализированной ЭВМ и особенности ее системы команд;

• ***алгоритмические языки,*** запись программ, на которых почти полностью унифицирована и производится с учетом лишь

структуры памяти, специализированной ЭВМ или класса ЭВМ, либо вообще без учета особенностей конкретной ЭВМ, предназначенной для работы по этой программе.

Для специализированных ЭВМ указанные языки должны были образовывать взаимосвязанную ***систему языков программирования,*** в которой обеспечивался преемственный ***вертикальный переход*** от нижних уровней на более высокий уровень путем ввода новых конструкций в язык и, наоборот, исключением конструкций, не соответствующих задачам уровня. Разработчик функциональной программы или модуля специализированной ЭВМ могли бы выбирать уровень языка в зависимости от требований к эффективности использования ресурсов памяти и производительности. Удовлетворение этих требований зависело от характеристик трансляторов.

Было исследовано расширение текста транслированных программ при программировании на языках высокого уровня. Для специализированных ЭВМ по отношению к программе, записанной на автокоде, трансляторы ***с макроязыка давали расширения не более 10 %, а с алгоритмического языка около 20 %.*** Поэтому при жестких требованиях к эффективности программы по занимаемому объему памяти и производительности, программирование рекомендовалось проводить на автокоде. При более слабых требованиях к полноте использования ресурсов специализированных ЭВМ и хороших трансляторах, алгоритмические языки оказывались предпочтительными, поскольку программирование на них удобнее и сопровождается меньшим числом программных ошибок.

Описание базового автокода представляло собой унифицированный документ, который использовался при программировании любой специализированной ЭВМ. Он содержал сведения о принципах построения базового автокода и структуре его документации. В особом разделе приводилась ***структура программ, правила оформления модулей, записи связей по управлению и кодированию программ.*** В описание базового автокода вводился специальный раздел, содержащий правила формирования конкретного представления автокода, которые регламентировал действия по обозначению машинно-зависимых элементов автокода. Все ***функции транслятора с автокода***, кроме генерации машинных команд, следовало адаптировать к явным значениям параметров специализированной ЭВМ, автокода и характеристикам проекта прикладной системы – заказа. Программы, реализующие перечисленные функции, следовало проектировать по методу

параметризации.

В большинстве проектов ***макрогенератор*** не зависел от параметров условий применения. Однако в специальных проектах в него могли быть встроены машинно-зависимые функции распределения памяти и масштабирования. В этом случае подготовка макрогенератора осуществлялась аналогично подготовке компилятора алгоритмического языка.

Компилятор алгоритмического языка необходимо было настраивать на структуру памяти и форматы специализированной ЭВМ, а также на правила программирования операторов алгоритмического языка, учитывающие специфику системы команд специализированной ЭВМ. Настройка на параметры машины могла осуществляться тем же методом, который применялся для настройки транслятора с автокода.

Адаптируемость интерпретатора для отладки программ специализированной ЭВМ ***с исполнением на универсальной ЭВМ,*** при выбранных показателях качества по производительности, обеспечивалась совместным применением принципов проектирования адаптируемых программ. Для параметризованной компоненты интерпретатора в базу данных для настройки включались информационные модули, содержащие всю информацию о структуре, специализированной ЭВМ.

В результате проведенных исследований были сформулированы основные ***особенности реализации системы автоматизации программирования и отладки,*** которые практически полностью были реализованы в системе ЯУЗА-6, построенной на технологической ЭВМ БЭСМ-6 в середине 1970-х годов. Исключение составляли задачи комплексной отладки управляющих программ в динамике, для реализации которых необходима была организация взаимодействия технологической и реальной управляющей ЭВМ.

Единство системы автоматизации программирования и отладки для всего технологического жизненного цикла комплексов программ обеспечивало разработку высококачественных программных продуктов при комплексной автоматизации всего технологического цикла и минимальных затратах совокупного труда на их создание. Система имела средства общения человека со средствами автоматизации на языках задания и программирования, построенных по единым правилам.

Настраиваемость системы автоматизации программирования и отладки на логику ***систем команд специализированных, мобильных ЭВМ*** была унифицирована и

сделана независимой от типа и системы команд ЭВМ. Вся специфика, обусловленная системой команд конкретной ЭВМ, была вынесена в сменные или настраиваемые программные и информационные модули. Настройка осуществлялась автоматизировано по формальному описанию логики команд и структуры мобильной ЭВМ, представленному на специальном языке.

Сменная (машинно-ориентированная) часть системы автоматизации программирования и отладки обеспечивала, с одной стороны, машинную ориентацию процесса трансляции с каждого из входных языков на язык системы команд конкретной специализированной ЭВМ, а с другой – позволяла исполнять программы этой ЭВМ на универсальной машине при отладке программ по тестам. При этом входные языки программирования, методы и правила структурного построения управляющих программ, язык отладки, состав и правила оформления документации на алгоритмы и программы являлись едиными для всех систем управления.

Обеспечивалось автоматизированное сопряжение отдельных модулей, компонентов и подпрограмм в большие комплексы управляющих программ. Для управляющих программ характерно мультипрограммное функционирование комплексов взаимодействующих подпрограмм объемом в десятки и сотни тысяч команд, использующих общую память глобальных переменных. ***Модульное, иерархическое построение*** позволяло получать обозримое описание решаемых задач и их взаимосвязей в сложном комплексе программ, а также существенно облегчать автоматизацию всех этапов процесса разработки и в первую очередь контроля компонентов и их сопряжения в единый комплекс.

Производился ***автоматизированный контроль корректности*** текстов, структуры компонентов программ и сопряжения всего комплекса программ. Во всех задачах контроля следовало иметь эталон – систему правил, которым должна соответствовать программа в целом и каждая ее часть. Структурный контроль построения компонентов выявлял некоторые виды зацикливания, наличие тупиковых и лишних участков в алгоритмах и другие нарушения правил построения топологической структуры подпрограмм.

Автоматизированная отладка автономных модулей и подпрограмм на уровне входного языка предназначена для локализации ошибок в системе, контролировать процесс вычислений при заданном тесте с различной степенью подробности. Значительная часть отладки при программировании

на алгоритмических языках, могла предварительно проводиться с использованием метода компиляции, т. е. по программам, представленным в командах технологической ЭВМ. Это позволяло существенно повысить скорость исполнения программ и снизить общую трудоемкость отладки. Исполнение отладочных заданий на заключительных стадиях отладки должно было производиться ***методом интерпретации на уровне команд специализированной ЭВМ.***

Комплексная отладка систем управляющих программ реального времени учитывала, что существует противоречие между стремлением обеспечить функционирование комплекса управляющих программ в условиях, максимально близких к реальным, и ограниченными возможностями управляющих ЭВМ для размещения и исполнения технологических программ, обслуживающих отладку. Кроме того, мобильные ЭВМ весьма ограничены по составу внешних устройств, необходимых для общения человека с машиной в процессе отладки комплекса программ в реальном времени.

Автоматизированный выпуск и корректировка технической документации были необходимы, для того чтобы успешно разработать, отладить и ввести в эксплуатацию сложный программный продукт. Документация на комплексы управляющих программ строилась по иерархическому принципу и состояла из нескольких уровней детализации. Такая структура должна позволять в удобной и наглядной форме проводить анализ программ как от общего к частному, так и от частного к общему. Наибольший объем документации соответствует представлению комплекса программ в наиболее детальном виде на уровне текстов программ на разных языках и описаний переменных и констант.

Автоматизированный контроль процесса разработки и технологических характеристик комплекса управляющих программ необходим руководителям проекта для учета объективных характеристик и тенденций изменения состояния производства продукта. В зависимости от степени детализации и целей анализа, эти характеристики должны служить основой для принятия различных решений по корректировке программ, технологии их разработки и распределению усилий специалистов с целью повышения качества проектирования и снижения его трудоемкости.

Структура, системы автоматизации программирования и отладки ЯУЗА-6 делилась на ***три крупные системы*** (см. рис. 3) [18]:

• организующую систему;

• систему автоматизации программирования;
• систему автоматизации отладки.

Эти три системы использовали развитую информационную систему – ***базу данных архивов,*** которые включали архив символьной информации исходных текстов программ, библиотеку паспортов подпрограмм, архив оттранслированных программ на языке программирования и в машинных кодах, тесты и результаты отладки, технологические и эксплуатационные документы программных продуктов.

Организующая система предназначена для формирования режимов функционирования САРПО, для управления хранением и обработкой данных, для подготовки системы к эксплуатации, а также для сбора и подготовки информации контроля процесса разработки.

Система автоматизации программирования обеспечивала получение синтаксически, семантически и структурно-корректной записи программ и описаний переменных, на входных языках САРПО ЯУЗА-6, получение машинных программ, а также технической документации на них и временных характеристик.

Система автоматизации отладки (система детерминированного тестирования) была предназначена для исполнения программ управляющей ЭВМ на технологической машине БЭСМ-6 с использованием программного интерпретатора, который моделировал работу специализированной ЭВМ.

При применении ЯУЗА-6 был налажен и апробирован ***процесс накопления и использования наборов готовых испытанных программных и информационных модулей и компонентов*** для формирования новых комплексов программ и/или их модернизированных версий. Библиотеки наборов таких компонентов и унифицированные межмодульные интерфейсы обеспечили ***эффективное конфигурационное управление*** при создании из них новых проектов и при модернизации эксплуатируемых версий программных продуктов. Такое комплексирование компонентов программ в крупные программные продукты позволило резко ускорить ряд разработок, повысить качество и снизить трудоемкость новых проектов программных продуктов ряда оборонных систем.

Полная версия САРПО ЯУЗА-6 эксплуатировалась в ряде организаций почти 20 лет с 1979-го года (в организации «Прогресс» до 2000-го года) и имела суммарный объем около 400 тыс. слов БЭСМ-6. В разработке этой версии системы принимали участие около 60 специалистов при средней производительности труда около 5 команд в день на человека. Процессы разработки компонентов системы производились сверху

вниз по техническим заданиям и спецификациям требований небольшими группами специалистов на языке БЕМШ. Регистрировались затраты на различных этапах, и произведена оценка общей трудоемкости разработки САРПО ЯУЗА-6, составившей около 300 человеко-лет. Для системы было выпущено техническое описание, инструкции по адаптации и эксплуатации.

Первая версия САРПО была ***передана для эксплуатации разработчикам в середине*** 1975-го года, настроенной на ЭВМ 5Э26 для разработки комплекса программ РЛУ «Основа», а затем системы

«Байкал». 5Э26, являлась мобильной, управляющей, многопроцессорной, высокопроизводительной вычислительной системой, построенной по модульному принципу (см. п. 2.6). К 1988-му году выпущено около 1,5 тысячи этих машин.

Наиболее активно САРПО ЯУЗА-6 применялась для различных типов ЭВМ «Аргон» (см. главу 2), а также в следующих организациях, где была ***адаптирована для приведенного количества типов специализированных, мобильных или бортовых ЭВМ:***

МНИИ приборной автоматики – 6 типов;

• НИИ автоматической аппаратуры – 5 типов;

• НПО автоматизации приборостроения – 11 типов;

• НИ электромашиностроительный институт – 2 типа;

• НПО ЭЛАС (г. Зеленоград) – 3 типа;

• НПО им. Лавочкина – 2 типа;

• НПО ПРОГРЕСС (г. Самара) и др.

В сумме это определило использование ЯУЗА-6 более чем в 13 организациях, для более 30 типов мобильных, специализированных ЭВМ. Общий ***объем разработанных программ с применением ЯУЗА-6 к*** 1985-му году ***превысил 5 млн. команд.*** В НПО АП ЯУЗА-6 использовалась, в частности, для разработки программ: орбитальной станции САЛЮТ-7; межпланетных станций Венера и Марс, спутников Экран, Радуга, Горизонт и ряда стратегических ракет. Эксплуатация ЯУЗА-6 постепенно прекращалась в конце 1990-е годов, в основном, вследствие технического старения машин БЭСМ-6.

3.6. История технологии программирования для бортовых ЭВМ реального времени на ЕС ЭВМ в 1980-е годы

В начале 1980-х годов стали доступными универсальные ЕС ЭВМ старших моделей с достаточными ресурсами для разработки и применения САРПО. Поэтому работы в МНИИПА сосредоточились на крупных системах ***РУЗА и ПРОТВА.*** Наибольшие функциональные возможности автоматизации жизненного цикла мобильных комплексов программ реального времени, были реализованы в системах ЯУЗА-6 и РУЗА. Объем оригинальных инструментальных программ в каждой из них составлял около 400 тыс. команд, а трудоемкость разработки около трехсот человеко-лет. ***Системы РУЗА и ЛУЗА-6*** функционально подобны, однако при создании системы РУЗА были учтены и устранены некоторые недостатки конструкции ЯУЗА-6, а также расширены функции системы в основном за счет ***компонентов программной инженерии.***

Особенности системы автоматизации разработки программ и отладки РУЗА (Александр Аркадьевич Штрик) [26]. Для построения системы взаимосвязанных языков программирования в системе РУЗА в качестве базового языка был принят процедурный язык высокого уровня. Языки РАДА являлись чистыми подмножествами языка АДА с точностью до русифицированной лексики (резервированные слова и предопределенные идентификаторы). Определены три версии языка программирования РАДА, совместимые снизу-вверх: РАДА-0, РАДА-1 и РАДА-2. Ядро языка (РАДА-0) было предназначено для разработки функциональных программ экстренного исполнения. В ядре языка реализованы: модульность, типизация данных (без ссылочных типов), структурное программирование. Модулями в языке РАДА-0 являлись процедуры, функции и пакеты данных, все модули являлись библиотечными.

Первое расширение (РАДА-1) было предназначено для разработки программ параллельного исполнения, главным образом программ организации вычислительного процесса и обмена, в режиме реального времени. РАДА-1 включала ядро РАДА-0, средства параллелизма (задачи) и исключительные ситуации. Второе расширение (РАДА-2) практически совпадало по

функциональным возможностям с полным языком АДА и использовалось для разработки основных настраиваемых компонентов программных средств (стандартных программ, программ ввода-вывода), а также функциональных компонентов с динамической организацией памяти.

Функционально САРПО РУЗА делилась на пять частей.

организующая часть для настройки на характеристики специализированной ЭВМ, условия применения и организации оперативной работы САРПО в соответствии с заданием пользователя;

• информационная часть, содержала средства централизованного управления базой данных проектирования и всю информацию о проектируемом комплексе программ и обеспечивала взаимодействие с ней пользователей;

• трансляционная часть, включала средства контроля исходных текстов программ на языках РУЗА, трансляцию, перевод в машинные коды и загрузку их в память специализированной ЭВМ, которая моделировалась в базе данных проектирования на технологической ЕС ЭВМ;

• отладочная часть, была предназначена для планирования и проведения автономной детерминированной отладки компонентов программ специализированной ЭВМ методом интерпретации их исполнения на ЕС ЭВМ в соответствии с заданием, формируемым пользователем на языке отладки;

• сервисно-информирующая часть, содержала средства выпуска документов, машинных носителей, сбора и отображения статистических данных по процессу разработки компонентов и комплекса программ.

Четкая структуризация компонентов по функциональному признаку, модульность их построения, наличие централизованной базы данных и стандартных средств взаимодействия с ней создали предпосылки для гибкого управления формированием версий технологической системы. Это позволило вводить в эксплуатацию отдельные функции, не ожидая полного завершения всех проектных работ по созданию системы. Поэтапный ввод компонентов системы РУЗА в эксплуатацию помимо технико-экономической целесообразности позволял оперативно и оптимально распределять ресурсы разработчиков САРПО.

Структурно система, РУЗА состояла из 568 программных и 260 информационных модулей. В среднем количество программных модулей в подсистемах составляло около 40. Основное количество модулей (82 %) было разработано на языке ПЛ1 (в среднем 130 операторов в модуле); Ассемблер

использован в тех случаях, когда необходимо обеспечивать связь с ОС ЕС, либо для создания отдельных высокоэффективных программ (в среднем 220 команд ассемблера). Из входивших в состав версий РУЗА подсистем, шесть являлись машинно-зависимыми; при этом перепрограммированию должны были подвергаться восемь программных модулей без учета разработки интерпретатора (для специализированной ЭВМ средней сложности это составляло 30–35 программных модулей) и автоматизировано изменяться 31 информационный модуль, что в сумме составляло примерно 10 % общего объема системы РУЗА. Система поставлялась с 1983-го года.

Система, автоматизации разработки программ и отладки ПРОТВА (Борис Аронович Позин) была предназначена для комплексной автоматизации процессов проектирования структуры, конструирования и производства программных средств реального времени, функционирующих на универсальных ЕС ЭВМ, а также на специализированных мобильных ЭВМ, программно совместимых с ЕС. В ***состав САРПО ПРОТВА входили.***

• инструментальные средства, обеспечивающие автоматизированное выполнение технологических операций в диалоговом и пакетном режимах;

• база данных проектирования, обеспечивающая накопление, корректировку и выдачу информации о комплексе программ и его составных частях на всех технологических этапах разработки;

• языки проектирования, описывающие комплекс программ и его составные части на всех технологических этапах;

• технологические инструктивно-методические материалы, включающие методики и правила разработки комплекса программ, разработки спецификаций требований, программирования, контроля, тестирования, комплексирования модулей.

Система позволяла создавать сложные комплексы программ размером до 1000 программных модулей (до нескольких сотен тысяч команд), функционирующих в реальном времени, разрабатываемых большими коллективами специалистов и эксплуатируемых много лет. САРПО ПРОТВА функционировала на универсальных ЕС ЭВМ моделях не ниже ЕС1033, имеющих в своем составе комплекс терминалов ЕС7906 или ЕС7920, под управлением операционной системы ОС ЕС версии 6.1, в режиме МУТ. Система поставлялась с 1984-го года.

3.7. Первая отечественная монография по проектированию крупных программных продуктов

Накопленный опыт создания программных продуктов в 1960-е – 70-е годы позволил выявить и сформулировать ***проблемы формирования и развития программной инженерии.*** Наметился перелом в ряде сфер от программирования и ***«художественного творчества»*** при индивидуальной разработке небольших программ к ***систематизированному, коллективному проектированию больших комплексов программ для систем управления и обработки информации в реальном времени.*** В 60-е годы происходил быстрый рост доступных ресурсов для реализации крупных программных продуктов, для различных областей применения ЭВМ и размеров комплексов программ. Резко повышались требований к качеству, ответственности и срокам создания крупных комплексов программ. Наиболее четко это проявилось ***в оборонных отраслях*** промышленности в стране и во всем мире. Тем самым эти отрасли стали ***«двигателями»*** развития и совершенствования ***программной инженерии.*** Проявилась необходимость осознать, сформулировать и опубликовать основные проблемы и характеристики развития этой индустрии, и ее роли в экономике страны. Для промышленного производства сложных программных средств следовало проанализировать недостатки применяемых технологий программирования небольших программ и определить основные направления их коренного совершенствования.

Одной из первых отечественных монографий, в которой обобщены проблемы программной инженерии, была опубликована В.В. Липаевым 1977-м году (18 тыс. экз.) [16]. В этой книге, по существу, сконцентрирована ***методология программной инженерии на уровне знаний и опыта некоторых предприятий оборонной промышленности страны в то время.*** Утверждалось, что низкое качество комплексов программ и низкая производительность труда алгоритмистов и программистов определялись в 60-е годы ***недостатками всего технологического процесса проектирования комплексов программ.***

В некорректной оценке и в неправильном распределении ресурсов на начальных этапах проектирования, прежде всего памяти и производительности ЭВМ, необходимых для решения

частных задач и общей целевой задачи создаваемой системы;

• в отсутствии или в недостаточном качестве планирования процессов разработки отдельных компонентов и всего комплекса программ;

• в плохом управлении коллективами специалистов и в слабом текущем контроле за состоянием комплекса программ, что объяснялось отсутствием методов, технических средств и возможностей для такого управления;

• в отсутствии достаточно определенной ответственности конкретных специалистов за разработку отдельных компонентов и всего комплекса программ на различных этапах проектирования;

• в низком уровне автоматизации технологического процесса проектирования и производства программ, в простоях алгоритмистов и программистов в ожидании результатов тестирования некоторых программных компонентов;

• в отсутствии четких критериев достижения успеха и завершения этапов разработки с оценкой количества и качества программного продукта и проделанной работы;

• в неравномерности реализации процессов жизненного цикла комплекса программ, с неторопливостью на начальных этапах и попыткой резкого форсирования работ на завершающих фазах.

Для их устранения необходимо было решать ***ряд общих проблем технологии проектирования комплексов программ,*** среди которых выделились [16]:

• организационные проблемы;
• методологические проблемы;
• структурные проблемы;
• технологические проблемы.

Организационные проблемы заключались, прежде всего, в создании методов и технологии эффективного и качественного производства комплексов программ различного назначения. Разработка и внедрение систем управления и информационно-справочных систем должны были оцениваться не только количеством систем, но и размером функционирующих в них различных программ. Программный продукт должен был выделен в дополнительную характеристику систем, которая отражала бы степень проведенной автоматизации. Полукустарное, дорогое и медленное создание комплексов программ должно было поднято на уровень автоматизированного производства сложных систем.

Важная организационная задача состояла также в разработке и внедрении стандартов и методик, определяющих обязанности заказчиков по ***формализации требований*** технических заданий и спецификаций на заказываемые системы управления и входящие

в них комплексы программ и их компоненты. Должны были созданы методы и средства, обеспечивающие разработчику и заказчику однозначное понимание характеристик систем и комплексов программ, подлежащих разработке, как в начале проектирования, так и при корректировках на последовательных фазах разработки.

Увеличение объема промышленного производства программного продукта требовало резкого ***увеличения выпуска программистов и системотехников*** вузами и техникумами. В то время вузы страны готовили специалистов по прикладной математике в количестве, не превышающем 5 тыс. человек в год, преимущественно для инженерно-исследовательских задач и информационно-справочных систем. Практически полностью отсутствовала подготовка системных аналитиков и математиков-программистов для проектирования и эксплуатации крупных программных систем управления объектами и технологическими процессами в реальном времени.

Метолологические проблемы состояли в исследовании характеристик процессов создания комплексов программ, и в выявлении этапов, в наибольшей степени влияющих на технико-экономические показатели программных продуктов. Необходимо было создать унифицированные методы измерения количества программного продукта и определения его качества, разработать методики расчета и прогнозирования трудоемкости и длительности разработки комплексов программ различного назначения с учетом их сложности, размеров, ограничений ресурсов, управляющих ЭВМ и других параметров. Особое внимание должно было обращено на создание ***методов оценки производительности труда*** коллективов, и отдельных разработчиков программ, с учетом этапов проектирования и неполной завершенности изготовления продуктов и их компонентов. Детальное исследование методических и программных ошибок, допускаемых при создании комплексов программ, должно было послужить основой для рационального распределения ресурсов при разработке технологий и инструментальных средств автоматизации проектирования.

Структурные проблемы заключались в создании унифицированных методов структурного проектирования сложных комплексов программ, эффективных по использованию ресурсов ЭВМ. Необходимо было разработать методы оптимального распределения ограниченных ресурсов ЭВМ в части использования внешней и оперативной памяти, производительности в реальном масштабе времени и взаимодействия компонентов в многомашинных и

многопроцессорных системах в зависимости от характера решаемых задач и параметров комплекса программ. Комплексы программ должны были строиться по модульно-иерархическому принципу с четкой унификацией правил оформления компонентов и их взаимного сопряжения (интерфейса) по управлению и по информации.

Увеличение размеров комплексов программ до десятков и сотен тысяч команд выдвинуло ***проблему надежности и безопасности*** применения таких комплексов при наличии искажений внешней информации, сбоев аппаратуры и не выявленных ошибок в программах. Структура комплексов программ непосредственно влияла на надежность функционирования и помехоустойчивость программ к различным возмущениям. Необходимо было исследовать факторы, влияющие на надежность функционирования крупных комплексов программ, и разработать методы создания структуры и компонентов программ, устойчивых к различным искажениям вычислительного процесса, а также методы оптимального распределения аппаратных и программных ресурсов ЭВМ, обеспечивающие безопасность программ.

Технологические проблемы включали создание унифицированной технологической схемы проектирования комплексов программ, четкого определения этапов производства, лиц, ответственных за их выполнение, и способов контроля количества и качества выполненной работы. Должны были предусмотрены методы и средства для корректировки и модернизации комплексов программ в течение всего времени их жизни и эксплуатации порядка 10–20 лет. Задача состояла в максимальной автоматизации единого технологического процесса ***с резким сокращением трудоемкого и рутинного, ручного труда*** при подготовке различного рода исходных данных, управляющих заданий, выходных документов.

Необходимо было разработать методы и программные средства для баз данных, обеспечивающих многолетнее надежное хранение и корректировку больших объемов информации, содержащих данные о разрабатываемых программных продуктах на различных уровнях описания: модели, алгоритмы, тексты программ на языках, машинные программы, испытательные модели, тесты, технологические и эксплуатационные документы.

Для получения программ ***с допустимым минимумом не выявленных ошибок*** следовало создать средства автоматизации автономной и комплексной отладки, унифицированные методы и средства для имитации информации внешних абонентов и для обработки результатов в процессе отладки и испытаний программ

динамического управления. Выходной программный продукт должен был снабжаться подробной документацией, характеризующей его количество и качество, позволяющей познавать, эксплуатировать, модернизировать и заменять управляющие программы частично или полностью без непосредственного участия специалистов, осуществивших их первичную разработку. Технология должна была поддерживаться группой связанных стандартов, методик и инструкций на технологические процессы жизненного цикла комплексов программ, на правила использования средств автоматизации, на типовую документацию, на правила контроля и приемки заказчиком готовых программных продуктов.

Эта книга имели хороший отклик в организациях оборонной промышленности и военных вузах. Материалы, изложенные в книге, были использованы профессором В.В. Липаевым в двухсеместровом курсе лекций «Технология программирования», которые читались в Московском инженерно-физическом институте на факультете «Кибернетика» в 1970 – м – 85-м годах. Однако в других вузах ***отсутствовал интерес и понимание проблем программной инженерии и создания крупных распределенных комплексов программ реального времени.*** Обучение студентов долго сводилось к программированию небольших относительно простых вычислительных программок. Также до конца 70-х годов очень слабыми были контакты со специалистами АН СССР и с вузами в значительной степени ***вследствие проблем секретности.*** В результате опыт и достижения организаций оборонной промышленности СССР по методам и технологиям разработки крупных комплексов программ реального времени сильно опережали (почти на десяток лет) академические разработки и, ***в то время не находили применения в гражданских отраслях народного хозяйства.***

Глава 4. История формирования основных компонентов программной инженерии в 1960-е – 70-е годы

4.1. Особенности крупных заказных функциональных задач для ЭВМ в 1950-е – 60-е годы

В 60-е годы ***заказчики оборонных систем*** непрерывно расширяли номенклатуру задач, возлагаемых на вычислительные средства, и повышали требования к качеству их решения. Этому сопутствовал быстрый рост объема и сложности комплексов программ, подлежащих реализации на ЭВМ. ***Принципиально изменялись объемы комплексов программ —*** происходил переход от индивидуального программирования ***«в малом»*** к индустриальному программированию ***«в большом»,*** то есть к коллективному созданию крупномасштабных программных продуктов высокого качества и надежности. Разработка таких комплексов программ в гражданских областях применения ЭВМ в эти годы не была востребована. Это определило ориентацию научных и учебных учреждений и предприятий гражданской промышленности на создание относительно небольших программ и ***способствовало их отставанию в индустрии проектирования крупных комплексов программ более чем на десяток лет***

Появление в конце 50-х годов ***принципиально нового вида промышленных изделий – программных продуктов,*** в которых сосредоточивалась основная, ***интеллектуальная сущность методов и процессов обработки информации и управления,*** а также значительная доля факторов, определяющих качество и эффективность систем в целом, для многих руководителей предприятий и проектов было непривычно и непонятно. Традиционно внимание руководителей проектов технических систем было сосредоточено на разработке аппаратуры – «железа», в том числе вычислительных средств. Видимая и осязаемая аппаратура ЭВМ, высокая стоимость ее разработки и производства создавали впечатление у многих руководителей проектов что, когда есть вычислительная техника – основное дело сделано, и далее предстоят небольшие затраты на программирование. Поэтому к разработке программ они первично не относились серьезно, почти не выделяли на это специалистов и время для проектирования, не оценивали необходимые затраты и ресурсы. Многие руководители считали, что сделать «бумагу» – программы можно без особой технологии и силами одного или нескольких «художников-умельцев». Труд программистов считался простым, дешевым и не требующим

специального технологического оснащения. Заказчики и руководители с удивлением и возмущением обнаруживали к концу разработки проекта, что комплексы программ не готовы и на их создание необходимого качества требуются затраты, соизмеримые с затратами на аппаратуру. Это зачастую приводило к острым конфликтам в коллективах разработчиков и с заказчиками, вследствие низкого качества или неготовности программных средств, что ***затягивало сроки завершения проектов систем*** в целом.

К концу 50-х годов в оборонных отраслях промышленности и в организациях министерства обороны страны быстро рос интерес к применению цифровых вычислительных машин для решения задач обработки информации и управления в системах военного назначения. Начались активные работы по освоению применения цифровой вычислительной техники для систем противовоздушной и противоракетной обороны, для контроля космического пространства и управления полетом в авиации и в космосе, для управления войсками и средствами вооружения разных видов. Многие из этих задач принципиально отличались по своему характеру, от ставших к тому времени традиционными, вычислительных задач в гражданских областях. В них преобладали логические операции, большая размерность, реальный масштаб времени и ряд других специфических свойств и требований. Очень быстро увеличивались номенклатура и объем функций систем, которые требовалось автоматизировать. Для реализации таких функций были необходимы значительные ресурсы памяти и производительности ЭВМ, а также большие коллективы специалистов, способные создавать крупные комплексы алгоритмов и программ в допустимые сроки. Уже первые комплексы программ военного назначения достигали нескольких десятков тысяч команд. В результате начало активно развиваться ***специфическое направление вычислительной техники для систем военного назначения*** [2, 3, 11].

Это направление развития ЭВМ почти одновременно начало формироваться в оборонных отраслях промышленности и на предприятиях, в нескольких проблемно-ориентированных областях для сухопутных, авиационных, морских, ракетных и других систем. Для последующего развития вычислительной техники, существенными оказались особые требования заказчиков различных областей применения. В результате, эти ЭВМ разделились на два класса: на ***стационарные,*** работающие в помещениях, и на ***мобильные***, размещаемые на подвижных (транспортабельные) или движущихся (бортовые), в том числе, необслуживаемых объектах. Этот фактор определил большие

принципиальные различия в архитектуре, технических, климатических и массогабаритных характеристиках этих ***двух классов специализированных ЭВМ военного назначения*** (см. главу 2). Первый класс тяготел к архитектурам и конструктивам стационарных, универсальных ЭВМ с необходимыми расширениями и модификациями для специализированного применения. Машины второго класса – мобильные, отличались наибольшей спецификой свойств задач и характеристик внешней среды применения, от остальных типов ЭВМ. При появлении цифровых вычислительных машин открылись широкие возможности для автоматизированного решения на них новых сложных вычислительных и логических задач обработки информации и управления, которые были недоступны ранее.

В начале 60-х годов начали промышленно производиться относительно небольшие, бортовые, мобильные и крупные территориально-распределенные ***вычислительные системы*** на базе средств телекоммуникации, функционирующие в реальном времени. Все эти работы проводились в режиме строгой секретности, и каждая функционально законченная система создавалась практически независимо, как от достижений за рубежом, так и от методов и результатов на других отечественных предприятиях.

Ориентация на решение конкретных функциональных задач и конструкционные требования минимизации энергопотребления, веса и габаритов, определяли ***предельно ограниченные ресурсы памяти и производительности мобильных ЭВМ.*** Отсутствие избыточности ресурсов заставляло разработчиков и заказчиков искать компромисс между широтой реализуемых функций, сложностью алгоритмов решаемых задач и необходимым качеством результатов функционирования системы. Ограничения ресурсов ЭВМ требовало от разработчиков экономного их использования и поиска архитектурных и технических возможностей совершенствования качества решения задач в пределах имеющихся ресурсов.

Жесткие ограничения, высокие и очень разнообразные требования к климатическим характеристикам и допустимым механическим перегрузкам, а также обычно очень высокие требования, к надежности функционирования принципиально не могли быть удовлетворены одним типом машин. В результате развивалась широкая гамма конструктивов для специализированных ЭВМ военного назначения. Перечисленные выше особенности функциональных задач отражались на архитектуре и структуре операций таких специализированных машин. Вследствие этого к концу 70-х годов ***сформировался***

очень широкий спектр (около 300) типов мобильных ЭВМ военного назначения, различающихся архитектурой и структурами команд, ориентированных на особенности функциональных задач, а также конструктивным оформлением, зависящим от областей применения. Эти ЭВМ отличались почти полным отсутствием вспомогательного и периферийного оборудования, не требующегося для непосредственного решения прямых функциональных задач при применении конкретной системы управления и обработки информации.

Развитие в те годы в стране технологии производства и элементной базы специализированных ЭВМ не поспевало за ростом требований к их ресурсам по памяти и производительности, необходимым для реализации всех новых требований и расширяющихся задач заказчиков. Одновременно очень быстро увеличивалась сложность и ответственность задач обработки информации и управления, возлагаемых на ЭВМ, что вызывало рост требований к качеству, надежности функционирования и безопасности применения комплексов программ для военных систем. При этом хронически не хватало вычислительных ресурсов для реализации всех новых и совершенствующихся задач в реальном времени. Для обеспечения решения этих сложных задач в очень ограниченных вычислительных ресурсах, ***архитектура, и системы команд специализированных ЭВМ приходилось тщательно адаптировать к характеристикам прикладных задач и сфер применения систем оборонного назначения.***

Систематически не выдерживались заданные сроки создания специальных вычислительных систем, прежде всего ***из-за потоков ошибок,*** выявляемых в программах, которые создавались первоначально в объектном коде без средств автоматизации программирования. Средняя производительность труда при разработке таких программ требуемого качества (по полному циклу разработки со сдачей заказчику) составляла всего 0,1–0,2 команды в день на человека, объем комплексов программ военного назначения уже тогда зачастую достигал и даже превышал 100 тысяч команд, а трудоемкость их создания составляла сотни человеко-лет. Отсутствие отработанной технологии, относительно низкая квалификация и оплата труда большинства программистов не стимулировали повышение производительности, высокое качество результатов программирования и систем в целом.

При разработке программ для первых распределенных вычислительных систем в начале 60 – х годов проявился «***заколдованный треугольник» —*** источники ряда ошибок могли

быть с одинаковой вероятностью в: аппаратуре – алгоритмах – программах. Все три источника в опытных образцах систем давали более или менее похожие внешние аномальные эффекты. В результате возникали ***острые конфликты между специалистами разных направлений,*** которые не желали признавать свои ошибки и требовались организационные меры и тщательные совместные исследования для определения их причин и источников. Особенно трудно приходилось разработчикам программ, результаты труда которых невозможно было просто «пощупать». Руководителям проектов систем и заказчикам представлялось, что после того, как изготовлена аппаратура – «железо», создать программы – «бумагу» не составляет особого труда, и они долго удивлялись и возмущались непрерывному потоку ошибок и низкому качеству решения функциональных задач.

Вследствие низкой производительности труда разработчиков программ в 60-е годы в мире провозгласили ***«кризис программирования»*** и острый недостаток специалистов для разработки программ. В результате приходилось интенсивно обучать новые кадры программистов и увеличивать их численность. Главным конструкторам систем и руководителям проектов таких программных комплексов стало ясно, что необходимо резко повышать производительность труда программистов и создавать ***новые автоматизированные технологии*** для разработки и обеспечения жизненного цикла комплексов программ управления и обработки информации военного назначения. Для этого были ***нужны новые принципиальные решения по методам, технологиям и инструментальным средствам для создания сложных программных продуктов.***

4.2. Особенности ЭВМ, ориентированных на задачи в реальном времени

Системы оборонного назначения с самого начала применения вычислительной техники отличались широким спектром принципиально новых и очень разнообразных, преимущественно логических, задач. По мере их апробации, количество и разнообразие задач возрастало катастрофически. Для их решения разрабатывались соответствующие новые алгоритмы, которые ***должны были ориентироваться на реальные условия применения и доступные ресурсы*** при реализации в соответствующих специализированных ЭВМ. Алгоритмы частных задач комплексировались в более сложные целостные системы алгоритмов, для координированного решения основных целевых задач соответствующих систем. Для этого необходимо было создание новых методов и алгоритмов объединения и комплексного взаимодействия разнообразных алгоритмов на единой или распределенной вычислительной среде в реальном времени.

Одна из важнейших особенностей систем оборонного назначения состояла в том, что в них большинство процессов протекает очень быстро и для принятия и реализации решений ***допустимое время реакции исчисляется секундами или даже долями секунды.*** Поэтому вычислительные средства должны были обеспечивать обработку информации и подготовку управляющих воздействий с высоким темпом, соответствующим динамическим процессам во внешней среде системы. Это определяло требования к ЭВМ и комплексам программ обеспечивать реализацию всех вычислительных процессов в регламенте ***жесткого реального времени*** и малого допустимого времени запаздывания реакции на поступающую внешнюю информацию. Эти требования должны были реализовываться с необходимым качеством при любых реальных потоках внешней информации, несмотря на ограниченные вычислительные ресурсы.

В программах специализированных ЭВМ оборонного назначения быстро проявилась принципиальная особенность, состоявшая в наличии ***двух классов переменных —*** непрерывных результатов измерения характеристик или пространственных координат и параметров движения внешних объектов, а также логических признаков свойств объектов и особенностей их функционирования. При этом в программах ***преобладали в большом количестве логические операции*** и относительно не

высокую долю составляли вычислительные процедуры. Анализ состава операций в программах уже первых систем военного назначения выявил, что через 5 – 10 команд следует условный переход и их относительное число составляет до 15–20 %, арифметических операций не более 10 %, а операции с индексными регистрами и пересылки между регистрами достигают 30–40 % от всех одноадресных команд. Подобные распределения доли различных типов операций ЭВМ впоследствии подтвердились в большинстве проектов комплексов программ оборонного назначения.

Значительная часть исходных данных в рассматриваемых системах являлась ***результатом квантованных измерений непрерывных физических величин –*** координат объектов и их скорости, показаний датчиков напряжения, давления и т. д. Относительно невысокая точность этих данных определяла рациональную разрядность квантования их значений для обработки и хранения на ЭВМ. Поэтому структура разрядной сетки и операции с частью слова выбирались в соответствии с числом разрядов наиболее часто используемых величин, что позволяло экономить объем аппаратуры. Эти же обстоятельства определили практическое отсутствие в специализированных ЭВМ операций с плавающей точкой. Операции с фиксированной точкой были более экономными по использованию оборудования, и вполне удовлетворяли разработчиков программ. В результате были созданы машины с различной базовой разрядностью памяти и основных операций – 12, 16, 18, 20, 24, и т. п. разрядов.

Логические операции в программах производились, как правило, с малоразрядными величинами или даже с отдельными битами. Для их выполнения целесообразно было ориентировать процессоры ЭВМ на удобную и быструю работу с величинами соответствующей структуры и вводить значительное количество специализированных логических операции с различными частями слова. Часть слова памяти (1–6 разрядов) отводилась для различных признаков логических величин. Такая ориентация конструкции машин на особенности основных, функциональных задач позволяла при относительно слабой элементной базе достигать необходимых характеристик по памяти и производительности ЭВМ. Поэтому многие машины только в конце 70-х годов начали приближаться к менее эффективной для конкретных приложений в данной области, универсальной, байтовской структуре операций и обрабатываемых данных.

Специализация систем команд ЭВМ для эффективного решения особых функциональных задач систем, усложняла труд

программистов, однако позволяли их решать при минимальных затратах на аппаратуру вычислительной техники. Кроме того, программирование могло начинаться только после полной готовности к применению всей аппаратуры. В результате в крупных системах разработка комплексов программ становилась определяющей совокупные затраты и сроки их производства.

Функциональные задачи многих оборонных систем управления в реальном времени характеризуются ***интенсивными случайными потоками неоднородных данных, длительность обработки которых являются также случайными величинами.*** Эти длительности во многом зависят от содержания поступающих данных и случайных моментов их обработки. Моменты и способы включения программ на обработку данных в специализированных ЭВМ производятся в основном автоматически и зависят от информации, передаваемой по телекоммуникационным линиям от операторов или из внешней среды. При этом, зачастую, нет необходимости при функционировании систем накапливать и хранить большие объемы данных продолжительное время. Поэтому базы данных о состоянии внешней среды отличались сравнительно небольшим объемом и практически ***отсутствовали специальные системы управления базами данных.*** Только некоторые стационарные военные системы (штабные) имели довольно крупные базы данных, но регламент реального времени у них был более легкий и допустимы большие времена задержки отклика.

Многие системы со специализированными ЭВМ предполагались для производства и применения в небольшом количестве (единицы, десятки, или сотни экземпляров), что ориентировало разработчиков на применение оригинальных технических решений и вызывало пренебрежение унификацией и стандартизацией аппаратуры, программ и технологии их производства. Практически независимая разработка такого ***широкого спектра вычислительных машин,*** конечно, обходилась очень дорого, однако в результате появлялось множество очень эффективных технических решений при разработке и применении ЭВМ и комплексов программ. Ведомства заказчиков оборонных систем не координировали между собой технические требования к вычислительным средствам, каждое из которых адаптировалось разработчиками к задачам конкретного заказчика. В результате ***бурно рос «зоопарк» типов вычислительных машин и программных комплексов,*** зачастую решающих одни и те же задачи. Особенно это проявлялось в многообразии машин и программных комплексов с подобными задачами в авиационных и ракетных системах.

Ориентация на решение конкретных функциональных задач и конструкционные требования минимизации энергопотребления, веса и габаритов, определяли ***предельно ограниченные ресурсы памяти и производительности многих ЭВМ.*** Отсутствие избыточности ресурсов ЭВМ заставляло разработчиков и заказчиков систем искать компромисс между широтой реализуемых функций, сложностью алгоритмов решаемых задач и необходимым качеством результатов функционирования системы. Ограниченность ресурсов ЭВМ требовало от разработчиков экономного их использования и поиска архитектурных, технических и алгоритмических возможностей совершенствования качества решения задач в пределах имеющихся ресурсов.

Реальных ресурсов, специализированных ЭВМ по памяти и производительности, обычно было недостаточно для полного и качественного решения даже основных, функциональных задач таких систем, и размещение на них инструментария автоматизации программирования было принципиально невозможно. Поэтому возникла необходимость ***распределять процессы создания комплексов программ*** оборонного назначения ***по двум классам ЭВМ.*** На универсальных ***технологических ЭВМ с*** большими ресурсами памяти и производительности реализовывать и применять инструментарий для автоматизации разработки программ, производить трансляцию программ с различных языков программирования, их предварительную отладку и документирование. На специализированных, ***объектных ЭВМ*** с ограниченными ресурсами, достаточными только для решения основных, функциональных задач системы, завершать их отладку и обеспечивать их эксплуатацию в соответствии с этими задачами. Эти ЭВМ обычно принципиально отличались от универсальных по архитектуре, системам команд, ресурсам и наличию периферии, что вызвало ***проблемы интерпретации программ специализированных ЭВМ на универсальных машинах,*** и моделирования на них тестов для имитации внешней среды при отладке и испытаниях (см. главу 2).

Отсутствие в стране в 1950-е – 60-е годы развитой централизованной промышленности электронных компонентов для ЭВМ, явилось причиной их разработки зачастую теми же предприятиями, которые создавали архитектуру специализированных ЭВМ и системы управления в целом. Вследствие этого ***элементная база часто была полукустарной, малотиражной и разнотипной,*** не отличалась высоким качеством и технологическим уровнем. Необходимость для многих предприятий оборонных отраслей вести разработку

систем по полному циклу, начиная с создания элементной базы ЭВМ и далее всей вычислительной техники и программных средств, не только приводило к множеству параллельных, не унифицированных разработок, но и значительно увеличивало длительность и стоимость проектов систем.

Только в начале 80-х годов в нашей стране начала проявляться ***тенденция частичной унификации архитектуры специализированных ЭВМ.*** Она была обусловлена необходимостью резкого ускорения и автоматизации разработки программ, для реализации инструментальных средств которой не было ресурсов памяти и производительности у объектных, специализированных машин. Их архитектуру стали ориентировать на наиболее распространенные в стране универсальные ЭВМ типа СМ или ЕС. Предполагалось, что это позволит полностью отрабатывать комплексы программ на универсальных ЭВМ, а затем переносить их без изменений на объектные ЭВМ.

Однако специфика ЭВМ военного назначения реального времени не полностью отражалась в универсальных ЭВМ, вследствие чего окончательную комплексную отладку и испытания приходилось проводить на реальных объектных ЭВМ. Этот процесс развивался очень медленно вследствие огромного объема ***«-унаследованных.»*** программ, уже используемых в реальных системах. Это позволяло сохранять испытанные и эксплуатируемые программы и вводить новые – без проведения полной разработки всего комплекса программ и системы управления с необходимыми испытаниями.

4.3. Создание операционных систем и глобальной телекоммуникационной сети ЭВМ реального времени в 1965-е годы

Интенсивное увеличение использования ресурсов ЭВМ потребовало автоматизировать регистрацию и упорядочивание запросов разных специалистов на решение их задач, и привело к созданию автоматизированных ***операционных систем распределения времени*** функционирования отдельных ЭВМ с учетом очередей пользователей и задач. Такие системы удовлетворяли индивидуальное использование ЭВМ ***независимыми специалистами*** в научных учреждениях и вузах при относительно небольшом спросе на их применение и небольших по размеру программах. Возрастание очередей на решение различных задач при ограниченном числе доступных ЭВМ в стране вызвало необходимость совершенствования механизмов управления последовательностью и эффективностью использования ресурсов отдельных автономных вычислительных машин.

Ниже ***история*** развития программной инженерии и телекоммуникационных сетей ЭВМ в нашей стране, в пятидесятые – шестидесятые годы прошлого века, рассматривается преимущественно на примере создания ***комплексов программ системы противовоздушной обороны (ПВО).*** У специалистов

отсутствовали знания и опыт создания сложных комплексов программ для цифровых вычислительных машин, методы решения задач обработки информации в реальном времени и организации телекоммуникационных сетей, взаимодействующих ЭВМ. Поэтому читателям следует учитывать, что изложенный ниже материал является в основном оригинальным, работы осуществлялись при очень скромных ресурсах вычислительных машин и слабых телекоммуникационных каналах связи. Автор монографии был участником этих работ, что возможно отразилось на субъективности некоторых концепций. Подобные работы проводились и на других оборонных предприятиях, но были строго засекречены и, недоступны для анализа (Рис. 4).

В 1959 году под руководством Анатолия Леонидовича Лившица и Залмана Михайловича Бененсона в НИИ—5 было проведено обобщение выполненных исследований, результаты которых отражены в большой комплексной работе по ***созданию территориальной информационной системы ПВО страны.***

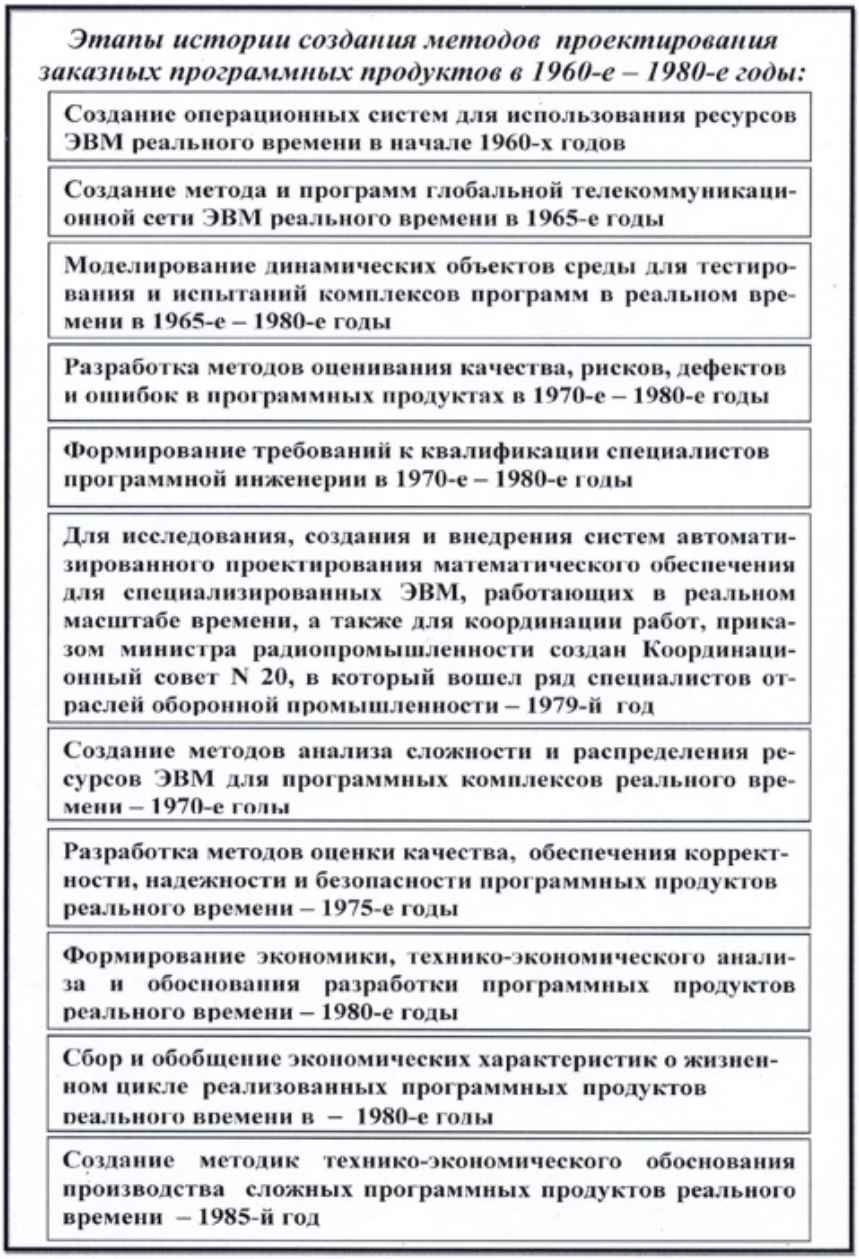
Этапы истории создания методов проектирования заказных программных продуктов в 1960-е – 1980-е годы:
Создание операционных систем для использования ресурсов ЭВМ реального времени в начале 1960-х годов
Создание метода и программ глобальной телекоммуникационной сети ЭВМ реального времени в 1965-е годы
Моделирование динамических объектов среды для тестирования и испытаний комплексов программ в реальном времени в 1965-е – 1980-е годы
Разработка методов оценивания качества, рисков, дефектов и ошибок в программных продуктах в 1970-е – 1980-е годы
Формирование требований к квалификации специалистов программной инженерии в 1970-е – 1980-е годы
Для исследования, создания и внедрения систем автоматизированного проектирования математического обеспечения для специализированных ЭВМ, работающих в реальном масштабе времени, а также для координации работ, приказом министра радиопромышленности создан Координационный совет N 20, в который вошел ряд специалистов отраслей оборонной промышленности – 1979-й год
Создание методов анализа сложности и распределения ресурсов ЭВМ для программных комплексов реального времени – 1970-е годы
Разработка методов оценки качества, обеспечения корректности, надежности и безопасности программных продуктов реального времени – 1975-е годы
Формирование экономики, технико-экономического анализа и обоснования разработки программных продуктов реального времени – 1980-е годы
Сбор и обобщение экономических характеристик о жизненном цикле реализованных программных продуктов реального времени в – 1980-е годы
Создание методик технико-экономического обоснования производства сложных программных продуктов реального времени – 1985-й год

Рис. 4.

Эта система должна была ***в реальном времени, объединять в глобальной сети ряд ЭВМ*** на узлах сбора и обработки радиолокационной информации о воздушных объектах, и на командных пунктах управления активными средствами ПВО. Все средства обработки информации и управления в системе должны были работать на ЭВМ в реальном масштабе времени, при

несинхронных потоках сообщений от удаленных ***независимых движущихся объектов —*** источников информации (самолетов и ракет). На каждом узле обработки радиолокационной информации ***(РЛУ)*** и командном пункте управления средствами ПВО должны были применяться, объединенные в локальную сеть графические терминалы различных типов для визуализации воздушной обстановки и обеспечения функционирования оперативного и командного состава, с временем отклика, измеряемого долями секунды. Система на вычислительных машинах должна была базироваться на совокупности транспортабельных ***РЛУ,*** создававших почти сплошное поле радиолокационного обнаружения различных динамических объектов во всем воздушном пространстве страны.

В результате в сферу исследований и разработок вошел ***новый широкий класс вычислительных систем,*** в которых основными компонентами и источниками информации являлись траектории динамических объектов, характеризующиеся их назначением, областью применения, координатами и обобщенными параметрами движения в реальном времени, определяющие требования к функциям комплексов программ. Следовало создать комплексы программ для ***обработки информации в реальном времени,*** соответствующие требованиям к характеристикам динамических объектов, их динамическим траекториям, и поставляющие обработанную информацию в системы управления активными средствами ПВО. Телекоммуникационные сети ЭВМ, должны были обеспечивать обмен и обобщение информации о динамических объектах и их траекториях на РЛУ для непрерывного сопровождения объектов.

В результате необходимо было решить следующие ***научно-технические задачи телекоммуникационной сети и РЛУ,*** и создать средства для их реализации:

- провести анализ и разработать принципиально ***новые динамические операционные системы реального времени*** на ЭВМ для синхронизации и управления решением разнородных задач;
- разработать ***комплексы функциональных алгоритмов и программ*** для обработки радиолокационной информации в реальном времени о движущихся динамических объектах;
- разработать ***телекоммуникационные сети*** для транспортировки информации между РЛУ о траекториях движения динамических объектов;
- разработать методы и средства для ***динамического тестирования и испытания корректности функционирования*** сложных программных комплексов обработки и транспортировки радиолокационной информации в реальном времени;

• ***связи*** программными комплексами обработки и транспортировки исследовать и обеспечить необходимую производительность и ***динамическое использование ресурсов ЭВМ и каналов*** радиолокационной информации в реальном времени.

В 1958-м – 59-м годах в НИИ —5 начались активные работы по применению ЭВМ для решения оперативных задач от нескольких несинхронных, независимых источников информации с учетом времени ее приема в РЛУ и командных пунктах системы. Эти источники различались важностью и длительностью решения их функциональных задач, величиной допустимого запаздывания выдачи потребителям результатов вычислений. Процессы решения задач должны были укладываться в определенные интервалы времени с учетом реальной производительности применяемых ЭВМ.

Функциональные задачи систем быстро росли по сложности, размеру программ, и потребной производительности вычислительных машин. Для динамического решения таких задач на предприятиях оборонной промышленности началась автоматизация управления их решением и разработка ***нового класса*** средств управления вычислительными процессами – ***создание операционных систем реального времени.*** Необходимо было оптимизировать использование производительности ЭВМ в сложных ситуациях несинхронных потоков информации из внешней среды о различных объектах, при разнообразии последовательностей и длительностей решения отдельных функциональных задач.

Первоначально, в конце 1950-х годов реальное время процессов в локальных системах обработки информации и управления могло иметь произвольную временную шкалу и обеспечивать только координацию решения функциональных задач конкретной ограниченной системы. По мере развития и расширения пространства наблюдения воздушных объектов в таких системах, как управление противовоздушной обороной, используемое в системах реальное время при решении функциональных задач должно было унифицировано и приведено к ***единой глобальной территориальной шкале*** на большом пространстве использования ряда РЛУ и ЭВМ телекоммуникационной сети ПВО. При обмене информацией между компонентами таких глобальных систем все сообщения о наблюдаемых объектах должны были снабжаться реальным временем в соответствии с единой шкалой. Несинхронный обмен радиолокационной информацией о движущихся воздушных объектах между различными удаленными ЭВМ, привел к

необходимости организации в них механизмов прерывания вычислений второстепенных задач для сохранения реального времени реализации основных, наиболее важных функций вычислительных процессов.

Значительные трудности реализации операционных систем реального времени в начале 1960-х годов состояли в ***неопределенности параметров*** временных динамических процессов поступления, длительности обработки и выдачи информации потребителям, а также в ограниченности производительности и ресурсов ЭВМ для размещения операционных программ. Все эти процессы решения функциональных и операционных задач управления должны были укладываться в те ограничения производительности ЭВМ, которые были ***доступны разработчикам в эти годы.*** К сожалению, в это время в отечественной и зарубежной литературе практически отсутствовали публикации по методам и организации операционных систем ЭВМ реального времени, все инженерные разработки и исследования в этой области проводились для конкретных проектов как секретные на соответствующих оборонных предприятиях.

Для эффективной реализации требований к функционированию системы вскоре были созданы и исследованы математические и статистические модели процессов конкретных операционных систем реального времени. Эти исследования базировались на достаточно развитой теории массового обслуживания и теории расписаний, применявшихся в различных инженерных областях, что позволило несколько улучшить качество функционирования операционных систем реального времени. В первых версиях операционных систем реального времени реализовывались бесприоритетные дисциплины, а затем они усовершенствовались и стали применяться дисциплины упорядочивания решения функциональных задач ***с учетом их относительных и абсолютных приоритетов.*** Такое построение операционных систем реального времени повышало эффективность использования ресурсов производительности и памяти ЭВМ на 10–15 %. Однако эти дисциплины не могли кардинально повысить производительность ЭВМ, часть низкоприоритетных задач могла пропускаться, сообщения терялись не использованными. Накопление и обобщение опыта и знаний о процессах и системах решения задач позволили сформулировать следующие ***требования к сложным операционным системам ЭВМ реального времени для автоматизации обработки информации*** о динамических объектах в системах:

• в процессах управления решением задач должно использоваться единое глобальное реальное время систем управления и обработки информации о динамических объектах, а также об изменениях внешней среды;

• потоки сообщений для обработки данных из внешней среды могут быть независимыми, несинхронными, различными по интенсивности, содержанию, реальному времени формирования и поступления для обработки;

• информация в сообщениях от источников и объектов внешней среды должна содержать реальное время, к которому относятся характеристики сообщений, их координаты и текущие состояния;

• реальное время решения различных функциональных задач должно эффективно упорядочиваться в соответствии с их приоритетами, установленной дисциплиной диспетчеризации, определенной при производстве системы, и реальным временем приема информации из внешней среды;

• алгоритмы и программы функциональных задач системы могут различаться по длительности решения и важности для пользователей, должны включать и использовать текущее и расчетное реальное время результатов решения и исходной информации;

• информация и сообщения для потребителей и внешней среды могут выдаваться асинхронно, в соответствии с установленной дисциплиной и содержать значения реального времени, которому соответствуют обработанные данные в сообщениях.

Полное отсутствие информации о подобных разработках систем в эти годы на Западе и на соседних оборонных предприятиях обеспечивало оригинальность технических решений, которые, в ряде случаев, впоследствии оказались ***новыми, принципиальными и достаточно универсальными для территориальных вычислительных систем реального времени различного назначения.***

В начале 1960-х годов был промоделирован, исследован, решен и практически апробирован ряд ***научно-технических задач создания программных средств реального времени*** для обработки радиолокационной информации на специализированных ЭВМ (В.В. Липаев, Л.А. Фидловский). Каждая радиолокационная станция имеет ограниченную зону наблюдения о воздушных объектах. Воздушный объект мог наблюдаться одновременно несколькими РЛС или ни одной. Для успешного управления активными средствами ПВО необходимо было обеспечивать возможности непрерывного использования

информации о таких объектах и формирования их траекторий движения в широком пространстве. Для этого данные о трассах воздушных объектов ***по телекодовым каналам связи*** обменивались между соседними РЛУ, и при последовательном их наблюдении ***координаты каждого объекта обобщались*** в единую трассу.

Один и тот же воздушный объект мог фиксироваться в пространстве в различные моменты реального времени, которое передавалось в составе сообщений на другие соседние РЛУ. Кроме того, сообщения могли задерживаться при ожидании в очереди для передачи в телекоммуникационные каналы связи и поступать для обобщения с измененным реальным временем. Таким образом, перед обобщением координаты и параметры движения объектов от различных источников для устранения в них различия реального времени необходимо было их пересчитывать (экстраполировать) ***к значению единого времени на момент обобщения координат объекта.***

Достоверность координат и параметров движения одного и того же объекта от различных РЛУ могла значительно различаться, что приходилось учитывать ***при их обобщении в единую траекторию.*** Для обеспечения взаимодействия РЛУ и обмена информацией между ними была создана ***телекоммуникационная сеть*** на базе специально выделенных каналов обычных телефонных линий связи. Структура сообщений в сети о наблюдаемых объектах и командах оперативного состава РЛУ была унифицирована, кодировалась и декодировалась в специальных устройствах, сопряженных с компьютерами РЛУ.

В сферу научных исследований и разработок в начале 60-е годов в Советском Союзе (почти одновременно в несколько ином виде в США) вошел и был апробирован ***новый широкий класс вычислительных систем и телекоммуникационных сетей реального времени – прототип*** современных информационных глобальных сетей и ***Интернета.*** В нем основными компонентами и источниками информации являлись траектории воздушных объектов, характеризующиеся их назначением, координатами и обобщенными параметрами движения, определяющие требования к функциям сложных комплексов программ управления.

4.4. Моделирование динамических объектов для тестирования и испытаний комплексов программ в реальном времени в 1960-е – 80-е годы

Сложные комплексы программ оборонного назначения должны обладать высоким качеством, для чего их необходимо тестировать и испытывать в условиях ***динамического воздействия информации от внешней среды*** максимально приближающейся к реальной. Однако создавать специальные полеты самолетов, имитирующих противника, истребителей-перехватчиков, пуски и полет ракет и другие действия средств реальной военной техники для отладки и первичных испытаний программ очень дорого, не рентабельно и опасно. При этом высокая стоимость и риск испытаний с реальными объектами почти всегда оправдывал значительные затраты на ***интегрированные имитационные системы внешней среды,*** если предстояли испытания критических программных продуктов с высокими требованиями к качеству функционирования программ, с длительным жизненным циклом и множеством развивающихся версий. Высокая сложность некоторых объектов внешней среды, особенно, если при их функционировании активно участвуют операторы-пользователи, не позволяла полностью автоматизировать всю имитацию тестовых данных, для крупных программных продуктов оборонного назначения. Поэтому при реализации интегрированных проблемно-ориентированных моделей приходилось использовать аппаратурные аналоги реальных объектов внешней среды для формирования части данных. Разумное ***сочетание реальных объектов внешней среды и программных имитаторов на ЭВМ*** обеспечивало создание высокоэффективных стендов с адекватными комплексными моделями объектов внешней среды, необходимыми для испытаний программных продуктов в реальном времени [17, 11].

В качестве альтернативы натурным испытаниям с реальными объектами уже в середине 60-х годов стали разрабатываться ***имитаторы – генераторы тестов, адекватные динамическому поведению реальных объектов внешней среды и средств вооружения.*** Одними из наиболее сложных и дорогих имитаторов внешней среды, применяемых для испытаний комплексов программ, стали использоваться модели и испытательные стенды: объектов систем противовоздушной

обороны, полета космических аппаратов; диспетчерских пунктов управления воздушным движением и другие. В частности, позже, благодаря отладке и испытаниям на имитационно-моделирующих стендах комплексов программ удалось совершить безупречный автоматический полет системы БУРАН. До настоящего времени обеспечивается надежное и устойчивое функционирование ряда орбитальных группировок космических аппаратов различного назначения, в том числе Системы Предупреждения о Ракетном Нападении (СПРН) [13, 14].

На полигоне в 1964-м году при натурных динамических испытаниях алгоритмов и комплексов программ обработки радиолокационной информации, ***впервые*** начало проверяться функционирование сложных комплексов программ ***реального времени в телекоммуникационной сети,*** тестирование и отладка которых требовала соответствующей специфической технологии. Это привело к ***принципиальному изменению методов тестирования,*** при которых динамика формировании и значения тестов движущихся объектов должны были по времени соответствовать текущим состояниям вычислений в проверяемых комплексах программ РЛУ. До этого ***в начале*** 1960-х годов для решения конкретных вычислительных задач, тесты обычно подготавливались независимо от процессов разработки проверяемых программ, и реальное время их подготовки не влияло на получаемые результаты вычислений. Однако для комплексов программ системы ПВО необходимо было их исполнение связать с реальным временем поступления и воздействием динамических тестов движущихся объектов, а также с текущим состоянием внешней среды.

В эти годы генерирование динамических тестов от внешних объектов ***на специализированных ЭВМ было невозможно*** вследствие ограниченности их вычислительных ресурсов, достаточных только для решения тестируемых функциональных задач, и потребовалось применять независимые большие универсальные ЭВМ, отделив во времени и в пространстве процессы формирования динамических тестов движущихся объектов, от моментов их использования. В 1965-м году для имитации тестов от движущихся объектов внешней среды ***в реальном времени*** были разработаны ***программы формирования магнито-фильмов*** на универсальной ЭВМ М-20 (Юрий Григорьевич Корольков). На этой машине предварительно формировались и записывались на специализированных магнитофонах ***наборы динамических тестов о разнообразных ситуациях воздушной обстановки и движения объектов, с регистрацией значений реального времени*** сообщений и их

координат. Для динамических траекторий воздушных объектов, можно было задавать различные маршруты и маневры в реальном времени. Это существенно ускорило подготовку тестов и динамическую отладку комплекса программ обработки радиолокационной информации в сложных ситуациях воздушной обстановки.

В конце 1960-х годов на основе накопленного опыта применения магнитофильмов для динамического тестирования и испытаний комплексов программ РЛУ была сформулирована концепция разработки моделирующего испытательного стенда (МИС) с перспективными функциями имитации динамики применения и развития всех компонентов внешней среды системы ПВО. Для его реализации были недостаточны ресурсы М-20 и требовалась более мощная универсальная ЭВМ, которая отсутствовала в институте.

В середине 1970-х годов началась разработка алгоритмов и программ ***стенда динамических испытаний, который был создан на БЭСМ-6 + АС-6.*** Стенд был способен достаточно полно заменить натурные испытания с реальными движущимися объектами внешней среды всей системы ПВО на полигоне и в НИИ—5. При этом высокая стоимость и риск испытаний с натурными объектами оправдывал значительные затраты ***на интегрированный стенд имитации тестов компонентов и фрагментов системы ПВО,*** если предстояли испытания критических программных комплексов с высокими требованиями к качеству их функционирования, с длительным жизненным циклом и множеством развивающихся версий.

Для сокращения неопределенностей и прямых ошибок при оценивании качества программ необходимо было до начала испытаний определить основные параметры внешней среды и потоки информации, при которых должен функционировать комплекс программ системы, с требуемыми характеристиками при оценивании его качества и эксплуатации. Для этого заказчик и разработчики совместно должны были структурировать, описать и согласовать ***модель динамики функционирования объектов внешней среды,*** их параметры в среднем, типовом режиме применения, а также в наиболее вероятных и критических режимах, в которых должны обеспечиваться требуемые характеристики качества динамического функционирования программ в реальном времени. К комплексам программ РЛУ и телекоммуникационных сетей заказчики системы могли предъявлять различные требования к надежности функционирования, к функциональной безопасности, к производительности и эффективности использования ресурсов

ЭВМ программным комплексом в реальном времени. Такая ***модель в реальном времени должна отражать характеристики.***

внешних, динамических потоков информации, о движущихся объектах внешней среды, их распределение по видам источников, характеристикам качества данных и возможным дефектам;

• интенсивность и структуру типовых сообщений от оперативных пользователей и администраторов сети, и их необходимую квалификацию, отражающуюся на вероятности ошибок и качестве выдаваемой информации;

• возможных негативных и несанкционированных воздействий от внешней среды при функционировании и эксплуатации программного комплекса ПВО;

• необходимые характеристики и ресурсы вычислительных средств, на которых предназначено функционировать комплексу программ испытываемой системы с требуемым качеством.

В отличие от натурных экспериментов, тестирование комплексов программ, моделирование внешней среды и динамических тестов движущихся объектов на ЭВМ, имеет большие возможности контроля, как исходных данных, так и всех промежуточных и выходных результатов функционирования испытываемого программного комплекса ПВО. В реальных сложных системах ряд компонентов иногда оказывается недоступным для контроля их состояния, так как-либо невозможно поместить измерители контролируемых сигналов в реальные подсистемы, подлежащие тестированию, либо это сопряжено с изменением характеристик самого анализируемого объекта. ***Программная имитация динамических тестов внешней среды на ЭВМ системы в реальном времени позволила:***

проводить длительное, непрерывное генерирование имитируемых данных для определения характеристик функционирования комплекса программ в широком диапазоне изменения условий и параметров экспериментов, что зачастую невозможно при использовании реальных объектов;

• расширять диапазоны характеристик имитируемых движущихся объектов, за пределы реально существующих или доступных натурных источников данных, а также генерировать динамические потоки информации, отражающие перспективные характеристики создаваемых систем и объектов внешней среды;

• создавать тестовые данные, соответствующие критическим или опасным ситуациям функционирования и движения объектов внешней среды, которые невозможно или рискованно реализовать при натурных экспериментах;

• обеспечивать высокую повторяемость имитируемых данных

при заданных условиях их генерации и возможность прекращения или приостановки имитации на любых фазах моделирования внешней среды.

Средства имитации внешней среды можно было разделить на ***две категории. Первая категория*** моделей используется при тестировании, квалификационных испытаниях программного продукта и не применялась при штатной эксплуатации программного продукта пользователями. ***Вторая категория*** более простых моделей внешней среды применялась непосредственными пользователями программного продукта для оперативной подготовки исходных данных при проверке различных режимов функционирования в процессе применения программ и при диагностике проявившихся дефектов. Такие модели ***в составе поставляемых заказчикам комплексов программ*** передавались пользователям для контроля функционирования их рабочих версий в реальном времени и входили в комплект поставки каждой пользовательской версии. Для размещения таких средств мониторинга и контроля качества комплекса программ необходимы были ресурсы внешней и оперативной памяти, а также дополнительная производительность ***объектной ЭВМ.***

До начала применения моделей внешней среды для испытаний программного продукта, они подлежали ***проверке и паспортизации,*** гарантирующей получение корректных эталонных данных и имитированных тестов, адекватных реальным внешним объектам. Для паспортизации моделей необходимы, прежде всего, достаточно полные и достоверные исходные характеристики реальных объектов внешней среды, которые должны были служить эталонами для этих моделей. Достигаемая достоверность имитации внешней среды, а, следовательно, и определения качества функционирования испытываемого программного продукта, естественно, зависели от ресурсов памяти, производительности и других характеристик ЭВМ, на которой реализуется модель.

Важнейшее значение для достоверности определения качества комплекса программ имеет ***адекватность имитаторов характеристикам объектов внешней среды,*** которая зависит от степени учета второстепенных факторов, отражающих функционирование реальных объектов или источников информации, при создании их моделей. Точность моделей на ЭВМ, прежде всего, определяется алгоритмами, на которых они базируются, и полнотой учета в них особенностей моделируемых объектов. Кроме того, на адекватность имитации влияло качество программирования и уровень дефектов и ошибок в программах

имитации.

В динамических системах и телекоммуникационных сетях ПВО, при излишне высокой интенсивности поступления исходных данных на обработку, может ***нарушаться временной баланс*** между длительностью решения требуемой совокупности задач программным комплексом ***в реальном времени,*** и ограниченной производительностью ЭВМ для решения задач. Важная задача испытаний компонентов и фрагментов системы ПВО состоит в определении рисков – вероятностей, с которыми будет нарушаться соответствие между потребностями в производительности для решения всей требуемой совокупности задач ***в реальном времени*** и возможностями ЭВМ и других компонентов системы. Однако если предварительно в процессе проектирования производительность системы в реальном времени не оценивалась или определялась слишком грубо, то, ***велик риск***, что доработки комплекса программ или фрагментов системы ПВО будут большими или может понадобиться заменить ЭВМ и компоненты телекоммуникационной сети, на более быстродействующие и с большей памятью. Это обусловлено, как правило, ***«оптимизмом»*** разработчиков, что приводит к ***занижению интуитивных оценок длительностей решения функциональных задач и*** возможных предельных интенсивностей потоков внешней информации в системе.

Для каждого эксперимента при испытаниях программ реального времени подготавливался план сценария тестирования и ***обобщенные исходные данные.*** Вызовы регистрирующих программ должны подчиняться определенной системе контроля динамического функционирования программ при исходной гипотезе, что ***некоторые ошибки и дефекты в программах и данных могут проявиться в любом компоненте и на любой стадии тестирования.*** Так как основная задача регистрации при тестировании в реальном времени состоит в обнаружении и локализации ошибок и причин отказов с точностью до функциональной группы программ или модуля, то более точное определение места дефекта следует переносить на тестирование в статике вне реального времени.

Испытания современных сложных систем обработки информации позволяют получать такое большое количество контрольных данных, что достаточно полный их анализ может представлять трудную методическую и техническую задачу, поэтому ***обработка результатов должна была осуществляться иерархически и дифференцировано.*** Для этого обработка результатов испытаний программ реального времени была разделена на две достаточно автономные части:

оперативную и обобщающую.

Так как качество функционирования оборонных систем обычно существенно зависит от характеристик конкретного человека, участвующего в обработке информации и управлении объектами, то необходимо было измерять эти характеристики и ***повышать путем обучения оперативного состава системы.*** Важной функцией моделей внешней среды является их использование в составе ***тренажеров для операторов-пользователей.*** Поэтому в процесс испытаний и эксплуатации органически входят процессы тренировки и изменения характеристик реальной реакции операторов, а также использование моделирующих стендов для обучения и регулярной подготовки операторов-пользователей для эксплуатации программных продуктов военного назначения. Кроме того, испытательный стенд может служить прототипом для разработки упрощенных тренажеров в составе систем обработки информации. ***Крупным достижением программной инженерии в*** 80-е годы ***в стране стало использование вы числительной техники для имитации внешней среды и динамических тестов систем в реальном времени.***

4.5. Методы оценивания качества, рисков, дефектов и ошибок в программных продуктах в 1960-е – 80-е годы

Источниками ошибок в комплексах программ являются специалисты – конкретные люди с их индивидуальными особенностями, квалификацией, талантом и опытом. В 1950-е – 60-е годы при индивидуальном создании относительно небольших программ, к ним обычно отсутствовали формализованные требования, и часто трудно было установить сам факт наличия ошибки. Не формализованные программистами функции и характеристики программ допускали неоднозначное толкование полученных результатов, в которых могли содержаться непредсказуемые дефекты или ошибки, закономерности которых было бесполезно обобщать. Однако в крупных комплексах программ в 1960-е – 80-е годы при наличии конкретных требований к программным продуктам, начинают изучаться статистика и распределение типов ошибок для коллективов разных специалистов, и проявляются достаточно общие закономерности, которые могут использоваться как ориентиры при их выявлении и систематизации. Этому могут помогать оценки типовых дефектов, модификаций и корректировок, путем их накопления и обобщения по опыту создания определенных классов программных продуктов на конкретных предприятиях. Актуальной становится формализация и прогнозирование рисков, дефектов и ошибок в определенных классах комплексов программ, что способствует достижению их высокого качества.

К ***понятию риски*** относятся негативные события и их величины, отражающие ***потери, убытки или ущерб для безопасности*** от процессов или продуктов, ***вызванные дефектами*** при проектировании требований, недостатками обоснования проектов, а также при последующих этапах разработки, реализации и всего жизненного цикла комплексов программ. Риски проявляются, как ***негативные последствия дефектов функционирования и применения программ,*** которые способны вызвать ущерб системе, внешней среде или пользователю, в результате отклонения характеристик объектов или процессов, от заданных требований заказчика, согласованных с разработчиками.

Оценки качества программных комплексов могут проводиться с двух позиций: с ***позиции положительной***

эффективности и непосредственной адекватности их характеристик назначению, целям создания и безопасности применения, а также с ***негативной позиции*** возможного при этом ущерба – риска от использования программного продукта или системы. Показатели качества преимущественно отражают положительный эффект от применения системы или программного продукта, и основная задача разработчиков состоит в обеспечении высоких значений качества. Риски характеризуют возможные ***негативные последствия дефектов*** или ущерб пользователей при применении и функционировании системы, и задача разработчиков сводится к сокращению дефектов и ликвидации рисков. Поэтому методы и системы управления качеством в жизненном цикле комплексов программ близки к методам анализа и управления рисками, они должны их дополнять и совместно способствовать совершенствованию программных продуктов и систем на их основе.

Характеристики дефектов и рисков непосредственно связаны с достигаемой ***корректностью, безопасностью и надежностью*** функционирования программных продуктов и ***помогают.***

оценивать реальное состояние проекта и прогнозировать необходимую трудоемкость и длительность для его положительного завершения;

• выбирать методы и средства автоматизации тестирования и отладки программ, адекватные текущему состоянию разработки и сопровождения, наиболее эффективные для устранения определенных видов дефектов и рисков;

• рассчитывать необходимую эффективность контрмер и дополнительных средств оперативной защиты от потенциальных дефектов и не выявленных ошибок;

• оценивать требующиеся ресурсы ЭВМ по расширению памяти и производительности, с учетом затрат на реализацию контрмер при модификации и устранении ошибок и рисков.

Риски могут быть обусловлены нарушениями технологий или ограничениями при использовании ресурсов – бюджета, планов, коллектива специалистов, инструментальных средств, выделенных на разработку. Результирующий ***ущерб*** в совокупности зависит от величины и вероятности проявления каждого негативного воздействия. Этот ущерб – риск характеризуется разнообразными метриками, зависящими от объектов анализа, и в некоторых случаях может измеряться прямыми материальными, информационными, функциональными потерями применяемых программных продуктов или систем. Одним из косвенных методов определения величины риска может

быть ***оценка совокупных затрат,*** необходимых для ликвидации негативных последствий в комплексе программ, системе или внешней среде, проявившихся в результате конкретного рискового события.

Процессы анализа и сокращения рисков должны сопутствовать основным этапам разработки и обеспечения ЖЦ сложных программных продуктов в соответствии с международными стандартами, а также методам систем обеспечения качества. ***Анализ и оценка рисков*** должны начинаться с исследования понятий, требований и функций, ***способствующих одобрению и применению*** заказчиком и пользователями конкретного программного продукта. При этом должны быть определены четкие требования к характеристикам и оценки возможного ущерба при их нарушении. Исследования процессов разработки комплексов программ в 80-е годы показали, что во многих случаях стоимость и длительность их реализации значительно превышали предполагаемые, а характеристики качества не соответствовали требуемым, что наносило ущерб заказчикам, пользователям и разработчикам. Эти потери – ущерб проектов, могли бы быть значительно сокращены своевременным анализом, прогнозированием и корректированием рисков возможного нарушения требований контрактов, технических заданий и спецификаций на характеристики, выделяемые ресурсы и технологию создания комплексов программ.

Управления рисками предполагает ясное понимание внутренних и внешних причин и реальных источников угроз, влияющих на качество продукта, которые могут привести к его провалу проекта или большому ущербу. В результате анализа следует создавать план ***отслеживания изменения рисков в жизненном цикле комплекса программ***., который должен регулярно рассматриваться и корректироваться. Главной целью управления рисками является обнаружение, идентификация и контроль за редко встречающимися ситуациями и факторами, которые приводят к негативным – рисковым результатам продукта. Это должно отражаться на применении регламентированных процессов, в которых факторы и угрозы рисков систематически идентифицируются, оцениваются и корректируются.

Понятие ошибки в программе — в общем случае под ошибкой подразумевается неправильность, погрешность или неумышленное искажение объекта или процесса, что может быть ***причиной ущерба – риска*** при функционировании и применении программы. При этом предполагается, что ***известно правильное, эталонное состояние объекта или процесса,*** по отношению к

которому может быть определено наличие отклонения – ошибки или дефекта. Исходными эталонами для любого программного продукта являются спецификации требований заказчика или потенциального пользователя, предъявляемых к программам. Любое отклонение результатов функционирования программы от предъявляемых к ней требований и сформированных по ним эталонов-тестов, следует квалифицировать как ***ошибку – дефект в программе,*** наносящий некоторый ущерб. Различие между ожидаемыми и полученными результатами функционирования программ могут быть следствием ошибок не только в созданных программах, но и ошибок в первичных требованиях спецификаций, явившихся базой при создании эталонов-тестов. Тем самым проявляется объективная реальность, заключающаяся в невозможности абсолютной корректности и полноты исходных требований и эталонов для сложных программных продуктов.

На практике в процессе ЖЦ исходные требования поэтапно уточняются, модифицируются, расширяются и детализируются по согласованию между заказчиком и разработчиком. Базой таких уточнений являются ***неформализованные представления и знания*** специалистов-заказчиков и разработчиков, а также результаты промежуточных этапов проектирования. Однако установить не корректность таких эталонов еще труднее, чем обнаружить дефекты в программах, так как принципиально отсутствуют формализованные данные, которые можно использовать как исходные. В процессе декомпозиции и верификации исходной требований, возможно появление ошибок в спецификациях на группы программ и на отдельные модули. Это способствует расширению спектра возможных дефектов и вызывает необходимость создания гаммы методов и средств тестирования для выявления некорректностей в спецификациях на компоненты разных уровней.

При тестировании обычно сначала обнаруживаются ***вторичные*** ошибки и ***риски,*** т. е. последствия и результаты проявления некоторых внутренних дефектов или некорректностей программ. Эти внутренние ***дефекты*** следует квалифицировать как ***первичные*** ошибки или причины обнаруженных аномалий результатов. Последующая локализация и корректировка таких первичных ошибок должна приводить к устранению ошибок, первоначально обнаруживаемых в результатах функционирования программ. Потери эффективности и риски программ за счет неполной корректности в первом приближении можно считать прямо пропорциональными (с коэффициентом) вторичным ошибкам в выходных результатах. Типичным является случай, когда одинаковые по величине и виду

вторичные ошибки в различных результирующих данных существенно различаются по своему воздействию на общую эффективность и риски применения комплекса программ.

Наибольшее число первичных ошибок вносится на этапах системного анализа и разработки модификаций программ. При этом на долю системного анализа приходятся наиболее сложные для обнаружения и устранения дефекты. На последующих этапах разработки комплексов программ ошибки вносятся и устраняются в программах в процессе их корректировки по результатам тестирования. Общие тенденции состоят в быстром росте затрат на выполнение каждого изменения на последовательных этапах процессов модификации программ. Интенсивность проявления и обнаружения вторичных ошибок наиболее велика на этапе активного тестирования и автономной отладки программных компонентов. Затем она снижается приблизительно экспоненциально. Различия интенсивностей ***устранения первичных ошибок, на основе их вторичных проявлений,*** и внесения первичных ошибок при корректировках программ, определяют скорость достижения заданного качества комплексов программ.

Одной из основных причин ошибок комплексов программ являются ***организационные дефекты при модификации и расширении их функций,*** которые отличаются от остальных типов и условно могут быть выделены как самостоятельные. Ошибки и дефекты данного типа появляются из-за недостаточного понимания коллективом специалистов технологии, а также вследствие отсутствия четкой его организации и поэтапного контроля качества продуктов и изменений. При отсутствии планомерной и методичной разработки и тестирования, остается не выявленным значительное количество ошибок, и, прежде всего, дефекты взаимодействия отдельных функциональных компонентов между собой и с внешней средой. Для сокращения этого типа массовых ошибок, активную роль должны играть лидеры – менеджеры и системотехники, способные вести контроль и конфигурационное управление требованиями, изменениями и развитием версий компонентов и программного продукта.

Первичные ошибки в программах можно анализировать с разной степенью детализации и в зависимости от различных факторов. Практический опыт показал, что ***наиболее существенными факторами, влияющими на характеристики обнаруживаемых ошибок, являются***:

• методология, технология и уровень автоматизации системного и структурного проектирования, а также

непосредственного программирования компонентов;

• длительность с начала процесса тестирования и текущий этап разработки или сопровождения, и модификации комплекса программ;

• класс программного продукта, масштаб (размер) и типы компонентов, в которых обнаруживаются ошибки;

• методы, виды и уровень автоматизации верификации и тестирования, их адекватность характеристикам компонентов и потенциально возможным в программах ошибкам.

Первичные ошибки в комплексах программ в порядке уменьшения их влияния на сложность обнаружения и масштабы корректировок можно разделить на следующие группы:

• ошибки вследствие большого масштаба – размера комплекса программ, а также высоких требований к его качеству;

• ошибки планирования и корректности требований модификаций часто могут быть наиболее критичным для общего успеха программного продукта и системы;

• системные ошибки, обусловленные отклонением функционирования реальной системы, и характеристик внешних объектов от предполагавшихся при проектировании;

• алгоритмические ошибки, связанные с неполным формированием необходимых условий решения и некорректной постановкой целей функциональных задач;

• ошибки реализации спецификаций требований – программные дефекты, возможно, ошибки нарушения содержания или структуры компонентов;

• программные ошибки, вследствие неправильной записи текстов программ на языке программирования и ошибок трансляции текстов программ в объектный код;

• ошибки в документации, которые наиболее легко обнаруживаются и в наименьшей степени влияют на функционирование и применение программного продукта.

Сложность проявления, обнаружения и устранения ошибок значительно конкретизируются и становятся измеримой, когда устанавливается связь этого понятия с конкретными ресурсами, необходимыми для решения соответствующей задачи и возможными проявлениями дефектов. При разработке и сопровождении программ основным лимитирующим ресурсом обычно являются допустимые трудозатраты специалистов, а также ограничения на сроки разработки, параметры ЭВМ, технологию проектирования корректировок. Показатели сложности при анализе можно разделить на ***две большие группы***:

• ***сложность ошибок при создании и корректировках*** компонентов и комплекса программ – статическая сложность,

когда реализуются его требуемые функции, вносятся основные дефекты и ошибки;

• ***сложность проявления ошибок функционирования*** программ и получения результатов – динамическая сложность, когда проявляются дефекты и ошибки, отражающиеся на функциональном назначении, рисках и качестве применения версии ПС.

Системные ошибки в крупных комплексах программ возникают, прежде всего, из-за неполной информации о реальных процессах, происходящих в управляемых объектах и внешних источниках информации. На начальных стадиях проектирования не всегда удается точно сформулировать целевую задачу всей системы, а также целевые задачи и характеристики основных функциональных компонентов, и их уточняют в процессе проектирования. В соответствии с этим уточняются и конкретизируются технические задания на отдельные функциональные программы и выявляются отклонения от требований уточненного задания, которые можно квалифицировать как системные ошибки. Ошибки, связанные с неполной формализацией целевой задачи системы, наиболее трудно было квалифицировать и выделять, в большинстве случаев они устранялись в процессе разработки программ.

Характеристики внешних объектов, принятые в процессе разработки алгоритмов за исходные, могут быть результатом аналитических расчетов, моделирования или исследования аналогичных систем. Во всех случаях может отсутствовать полная адекватность предполагаемых и реальных характеристик, что и являлось причиной сложных и трудно обнаруживаемых системных ошибок. Ситуация с системными ошибками дополнительно усложняется тем, что эксперименты по проверке взаимодействия программного продукта с реальной средой во всей области изменения параметров зачастую сложны и дороги, а в отдельных случаях, при создании опасных ситуаций, недопустимы.

Длительность отладки, а, следовательно, и всей разработки непосредственно зависит от допустимого значения показателя отлаженности или от количества прогнозируемых не выявленных ошибок, при котором разработку можно было считать завершенной. Однако в сложных комплексах программ детерминированный подход перестает быть конструктивным, и приходится переходить к ***статистической оценке уровня не выявленных ошибок и отлаженности комплекса.*** В этом случае показателем отлаженности может быть вероятность обнаружения ошибок в программе в течение некоторого времени или

интенсивность потока искажений результатов в период эксплуатации системы за счет не выявленных ошибок. В серийно выпускаемых программных продуктах реального времени, количество ежегодно обнаруживаемых ошибок сравнительно мало изменяется, что в значительной степени объясняется увеличением количества функционирующих экземпляров объектных ЭВМ и систем.

Взаимосвязь между количеством ошибок в программе, количеством выявляемых ошибок и интенсивностью искажений результатов может являться основой оптимизации суммарных затрат на оперативную защиту и отладку. Установлено, что для этого может использоваться жесткая корреляция между ***тремя вилами проявления ошибок*** в сложных комплексах программ [19]:

• суммарным количеством ошибок в комплексе программ или количеством неверных команд в программе по отношению к общему количеству команд;

• количеством ошибок в комплексе программ, выявленных в единицу времени в процессе отладки при постоянных усилиях;

• количеством искажений результатов на выходе комплекса программ вследствие не выявленных ошибок в программах.

Эти три показателя можно связать некоторыми постоянными коэффициентами пропорциональности. Теоретические исследования подтвердили эти связи и позволили создать методы ***статистического прогнозирования интегрального числа ошибок*** в зависимости от времени отладки комплекса программ. Созданная методика и проведенные исследования характеристик комплексов программ показали, что их целесообразно ***применять как ориентиры возможных ошибок и дефектов при разработке и сопровождении крупных программных продуктов.***

В 80-е годы при исследованиях за рубежом \\\]***примеров*** 20 крупных поставляемых программных продуктов, созданных в 13 различных организациях, коллективы специалистов добились среднего уровня 0,06 дефекта на тысячу строк нового и измененного программного кода. При использовании структурного метода в пяти проектах достигнуто 0,04 – 0,075 ошибок на тысячу строк. Таким образом, ***уровень ошибок около 0,05 на тысячу строк кола*** в разных публикациях считалось близким к предельному для высококачественных программных продуктов.

Другим ***примером оценок уровня ошибок*** особенно высокого качества может служить критический программный продукт бортовых систем Шаттла, созданный NASA. По оценке авторов, в нем содержится менее одной ошибки на 10000 строк кода.

Однако стоимость программного продукта достигает 1000 $ за строку кода, что в среднем в сто раз больше, чем для административных систем и в десять раз больше, чем – для ряда ординарных критических управляющих систем реально времени [11].

Приведенные характеристики типов дефектов и количественные данные могут служить ***ориентирами при прогнозировании возможного наличия не выявленных ошибок*** различных сложных программных продуктов высокого качества. Следующим логическим шагом процесса их оценивания может быть усреднение для большого числа проектов фактических данных о количестве ошибок на конкретном предприятии, приходящихся на тысячу строк кода, которые обнаружены в различных продуктах. Тогда в следующем проекте будет иметься возможность использования этих данных, в качестве меры количества ошибок, обнаружение которых следует ожидать при выполнении проекта с таким же уровнем качества, или с целью повышения производительности при разработке для оценки момента прекращения дальнейшего тестирования. Подобные оценки гарантируют ***от избыточного оптимизма*** при определении сроков и при разработке графиков разработки, сопровождения и реализации программ с заданным качеством. ***Непредсказуемость конкретных ошибок*** в программах приводит к целесообразности последовательного, методичного фиксирования и анализа возможности проявления любого типа дефектов и необходимости их исключения на наиболее ранних этапах ЖЦ при минимальных затратах.

4.6. Формирование профессий и квалификаций специалистов программной инженерии в 1970-е – 80-е годы

В эти годы очень быстро увеличивалась сложность и ответственность отдельных задач, связанных с обработкой информации и управлением, которые возлагаются на ЭВМ, что вызывает ***рост требований к качеству,*** надежности функционирования и ***безопасности применения*** программных продуктов. Индивидуальное программирование, при котором заказчиком, разработчиком, оценщиком и пользователем относительно небольших программ ***был один специалист,*** не могло справиться с резко возраставшими требованиями к размеру и качеству комплексов программ. В 50-е годы при программировании в кодах ЭВМ начали выделяться ***программисты – кодировщики и алгоритмисты,*** формулировавшие задачи на естественном языке или в виде графических блок-схем. По мере развития ассемблеров и алгоритмических языков в 60-е годы и возрастания требований к качеству программ появилась необходимость в использования независимых ***тестировщиков, а затем и сертификаторов*** программных продуктов. Приблизительно в это же время с увеличением размеров комплексов программ появились ***руководители-менеджеры и документаторы*** таких крупных продуктов и их

компонентов. Необходимым становилось структурирование и создание коллективов специалистов разных профессий и квалификаций, разделение профессионального труда под руководством менеджеров. В 80-е годы оформились более или менее определенные рациональные модели жизненного цикла проектирования и производства сложных программных продуктов требуемого качества, структуры и ***состав специалистов в коллективах.***

Исторически в 1950-е – 80-е годы ***принципиально изменились и возросли требования к их профессиональной квалификации, появляется необходимость обучения специалистов,*** связанных с созданием крупных программных продуктов, ряду важных ***для программной инженерии профессий.***

организации и регламентированной работе больших профессиональных коллективов специалистов над целостным продуктом;

• распределению сотрудников разной профессиональной

специализации по производственным этапам, компонентам и видам работ, в жизненном цикле комплексов программ;

• планированию и методам работы в условиях ограниченных ресурсов, по графикам в реальном времени, с заданными сроками, с поэтапным контролем качества и документированием результатов;

• тестированию, испытаниям и обеспечению гарантии качества, надежности и безопасности компонентов и программного продукта в целом.

Эти специалисты должны владеть ***новыми интеллектуальными профессиями.*** Они призваны обеспечивать высокое качество программных продуктов, а также контроль, испытания и достоверность реально достигнутого качества на каждом этапе разработки и совершенствования комплексов программ. Накопленный опыт создания крупных программных систем и острый дефицит, востребованных для выполнения таких работ специалистов, привели в 70-е годы к необходимости ***принципиального изменения и расширения методов и программ их обучения, и воспитания.*** Крупномасштабное производство программных продуктов различных классов, ***разделение труда специалистов по профессиональной квалификации*** при разработке программ, структура и организация больших коллективов, а также экономическая сторона таких производств, стали важнейшей частью ***процессов выбора, обучения и стимулирования специалистов*** для обеспечения всех этапов жизненного цикла сложных программных продуктов.

Создание программных комплексов, как производственной продукции, существенно повысило ***актуальность обоснования, прогнозирования и оценивания роли человеческого фактора*** для качества процессов производства. Технологии регламентированного проектирования и производства крупных программных продуктов большими коллективами специалистов ***принципиально отличались от технологий индивидуальной разработки небольших программ или комплексов программ свободным методом.***

руководители больших коллективов специалистов должны выполнять ***роль лидеров,*** объединяющих и координирующих знания, навыки и труд над программным продуктом, специалистов с разной профессиональной квалификацией и психологическими характеристиками;

• взаимодействие специалистов, ***творческая и психологическая совместимость*** в коллективе должны обеспечивать планируемое производство целостного программного продукта в реальном времени в заданные сроки и

требуемого качества;

• при формировании коллектива и выполнении совместных работ необходимо учитывать и использовать ***особенности каждого специалиста*** в коллективе, которые отличаются профессиональной квалификацией и психологическими характеристиками;

• следовало учитывать, что, как правило, сложно выделить ***персональное авторство и ответственность*** за реализацию отдельных функций и/или фрагментов, определяющих характеристики, качество, дефекты и риски компонентов и всего программного продукта;

• качество, поставляемого программного продукта зависит от ***качества труда почти каждого специалиста*** и его персональной квалификации, однако не всегда можно выделить конкретного специалиста, ответственного за выявленные критические дефекты, ошибки и риски неблагоприятных событий при применении программного продукта.

Как следствие роста сфер применения и ответственности функций, выполняемых программами, резко возросла необходимость ***гарантировать высокое качество программных продуктов,*** потребность регламентировать и корректно формировать требования к характеристикам подлежащих разработке компонентов и комплексов программ, к их реализации и проверке достоверности выполнения требований.

Лидеры – руководители разработчиков, участвующих в производстве крупных программных продуктов и высококвалифицированные специалисты в составе таких групп ***играют особую роль*** при формировании полноценных, корректных ***требований,*** которое должно осуществляться совместно с заказчиком и другими заинтересованными лицами, с участием экспертов по тематике области назначения продукта. ***Лидер должен был иметь талант и высокий уровень квалификации, а также иметь навыки,*** позволяющие ему:

• руководить процессом выявления, конкретизации и формирования требований заказчика проекта;

• осуществлять проверку спецификаций программного средства, чтобы удостовериться, что они соответствуют реальной концепции, представленной детальными функциями;

• квалифицированно вести переговоры с представителями заказчика, с пользователями и разработчиками, определять и поддерживать должный баланс между запросами заказчика, и возможностями команды разработчиков выполнить проект, используя выделенные заказчиком ресурсы, в течении запланированного на реализацию проекта времени;

• рассматривать «конфликтующие» пожелания, поступающие от различных участников проекта и находить компромиссы, необходимые для определения набора функций, которые в наибольшей степени удовлетворяют представителей всех сторон, заинтересованных в успешном выполнении проекта.

Чтобы добиться успеха в большом проекте, ***необходима четкая координация действий членов «команды»,*** которая должна работать по общей, заранее принятой методологии, чтобы реализовать комплекс требований и обеспечить качество программного продукта эффективно организованной командой разработчиков. При этом одним из наиболее важных факторов является то, что члены ***«команды»»*** имеют ***различные талант, профессиональные навыки и квалификацию.*** При выборе заказчиком надежного поставщика-разработчика проекта была необходима ***оценка тематической и технологической квалификации*** предполагаемого коллектива специалистов, а также его способности реализовать проект с заданными требованиями и показателями качества. Тематическую квалификацию специалистов в области создания программных продуктов определенного функционального назначения в первом приближении можно характеризовать средней продолжительностью работы основной части команды исполнителей в данной ***проблемной области***, непосредственно участвующей в разработке алгоритмов, спецификаций требований, программ и баз данных. Технологическая квалификация коллектива характеризуется опытом и длительностью работы с регламентированными технологиями, инструментальными комплексами автоматизации программной инженерии, языками проектирования, программирования и тестирования. Особенно важны, не столько индивидуальные характеристики каждого специалиста, сколько интегральный показатель ***квалификации «команды»,*** реализующей некоторую, достаточно крупную функциональную задачу или весь проект.

При создании высококачественных комплексов программ ***необходима организация и тесное взаимодействие представителей заказчика и руководителей*** проекта. Взгляды и требования заказчика, в основном, отражаются в функциональных и потребительских характеристиках версий программного продукта. Устремления разработчиков направлены на возможность и способы их реализации с требуемым качеством. Эти различия исходных точек зрения на проект приводят к тому, что некоторые неформализованные представления тех и других имеют ***зоны неоднозначности и взаимного непонимания требований*** к продукту, что может приводить к конфликтам при

выборе квалифицированного коллектива специалистов.

Экономическое обоснование проектов квалифицированными заказчиками и производителями на начальном этапе их проектирования должно содержать оценки рисков реализации поставленных целей, обеспечивать возможность планирования и выполнения жизненного цикла программного продукта или указывать на недопустимо высокий риск его реализации и целесообразность прекращения разработки. Большую часть рисков и негативных последствий производства можно избежать, освоив и используя существующие, ***методы оценивания и прогнозирования производственных затрат,*** а также управления проектами программных продуктов, для их успешного завершения.

Глава 5. История развития методов программной инженерии для систем реального времени в 1970-е – 80-е годы

5.1. Анализ сложности и распределения ресурсов ЭВМ для программных комплексов реального времени в 1970-е годы

Для решения проблем, представленных в главе 4, в апреле 1979 года министерство радиопромышленности СССР задало и обеспечило финансирование в МНИИПА (НИИ-5) отраслевой научно-исследовательской работы (НИР) ПРОМЕТЕЙ (Проектирование Математики Единая Технология). Это имя было присвоено всей последующей технологии и комплексу инструментальных средств на различных ЭВМ. Цель работы состояла в ***исследовании, создании и внедрении систем автоматизированного проектирования математического обеспечения для специализированных ЭВМ, работающих в реальном масштабе времени.*** Для координации работ коллективов разных предприятий в этой области, приказом министра в 1979 году был создан отраслевой Координационный совет № 20 министерства, в который вошли также некоторые специалисты других отраслей оборонной промышленности. Председателем совета, главным конструктором комплекса инструментальных систем и ПРОМЕТЕЙ-технологии в отрасли, был назначен профессор Владимир Васильевич Липаев. Для ***организации и выполнения этих работ*** были существенно увеличены коллективы специалистов, к началу 80-х годов в МНИИПА трудились около 200 специалистов и еще столько же участвовали в этих работах в качестве субподрядчиков в других предприятиях оборонной промышленности. Эти работы проводились по ***комплексной программе*** почти 10 лет до 1988 года. Общая трудоемкость работ по этой программе к концу 80-х годов составила около четырех – пяти тысяч человеко-лет [12, 11].

Программой работ предусматривалось выполнение научных исследований методов и процессов обеспечения жизненного цикла крупных комплексов программ реального времени, а также разработка и внедрение технологий и инструментальных ***систем, обеспечивающих.***

снижение стоимости, общей трудоемкости и длительности создания сложных комплексов программ для ЭВМ реального времени, и повышение производительности труда всех специалистов в коллективах разработчиков оборонных

программных продуктов;

• высокое качество, надежность и безопасность функционирования, создаваемых и сопровождаемых комплексов программ реального времени оборонного назначения в течение всего жизненного цикла;

• комплексную автоматизацию процессов коллективной разработки и модификации комплексов программ большого размера и высокой сложности;

• экономически эффективную, унифицированную технологию и инструментальные средства разработки и сопровождения комплексов программ различных ЭВМ широкого класса;

• эффективное использование ресурсов памяти и производительности ЭВМ сложными комплексами программ реального времени.

Разработке технологии в 1970-е – 80-е годы сопутствовало создание, анализ и научные ***исследования методов программной инженерии,*** а также обеспечения качества жизненного цикла крупных комплексов программ реального времени [17–20]. Практическое отсутствие готовых решений, и скудность публикаций в этой области в отечественной и зарубежной литературе стимулировали интенсивные исследования, и поиск эффективных, оригинальных методов разработки программных продуктов реального времени. Такие исследования проводились по многим направлениям, ***способствовавшим совершенствованию*** технологий и повышению качества создаваемых программных продуктов. Эти исследования органически входили в плановые работы по созданию инструментальных систем программной инженерии и финансировались в составе комплексной НИР ПРОМЕТЕЙ.

Интерес к ***определению сложности программных комплексов*** и выявлению характерных особенностей этого понятия первоначально в 60-е годы был обусловлен ***экспоненциальным ростом длительности решения ряда вычислительных задач*** в зависимости от количества переменных. Даже при сравнительно небольшом количестве переменных, характерном для типовых практических задач, оказывалось принципиально невозможным получать точные результаты на машинах в то время, ***путем перебора вариантов*** решения. При этом прогресс вычислительной техники хотя и позволял увеличивать размерность решаемых задач, однако получающийся прирост размерности не способен был снять принципиальные ограничения и удовлетворить требования пользователей. Были выявлены многие типы математических задач, которые трудно или невозможно было решать точно, что способствовало

развитию анализа сложности программ и вычислений на ЭВМ.

Понятие сложности интуитивно ассоциировалось с ресурсами, необходимыми для решения задач. Весьма расплывчатое понятие ***сложность комплексов программ*** значительно конкретизировалось и становилось измеримым, когда устанавливалась связь этого понятия с конкретными ресурсами, необходимыми для решения задачи. Сложность программ для систем реального времени преимущественно определялись допустимым временем отклика, а для информационно-поисковых систем – количеством типов обрабатываемых переменных. При проектировании программ основным лимитирующим ресурсом для создания любых продуктов обычно являлись ***допустимые трудозатраты специалистов*** при ограничении на сроки разработки, параметры ЭВМ, технологию разработки. Для этапа эксплуатации комплексов программ основными определяющими ресурсами становятся объем памяти команд и производительность ЭВМ, а в качестве ограничений могли выступать время счета, размер памяти переменных.

К группе параметров, влияющих на ***сложность разработки*** комплексов программ, относятся:

• величина программы, выраженная числом строк, команд или количеством программных модулей в комплексе;

• количество обрабатываемых переменных или объем памяти для размещения базы данных;

• трудоемкость разработки комплекса программ;

• длительность разработки;

• количество специалистов, участвующих в создании комплекса программ.

В 70-е годы стало ясно, что, варьируя требования к функциональным характеристикам программ, можно в значительных пределах изменять сложность и длительность их создания. Вычислительная сложность была непосредственно связана с ресурсами вычислительной системы, необходимыми для получения совокупности законченных результатов, и могла быть представлена тремя компонентами:

• ***временная сложность*** отражала необходимую длительность исполнения комплекса программ или время обработки на ЭВМ совокупности исходных данных до получения требуемых результатов;

• ***программная сложность*** характеризовалась длиной текста программы или размером памяти ЭВМ, необходимой для размещения программного комплекса;

• ***информационную сложность*** рекомендовалось

представлять, как объем базы данных, обрабатываемых комплексом программ, или как емкость оперативной и внешней памяти, используемой для накопления и хранения информации при исполнении программного продукта.

Эти параметры сложности программ при эксплуатации связаны между собой, и увеличение сложности по одному из показателей в некоторых пределах позволяет снижать сложность по другому показателю. Наиболее характерна такая ***связь между тремя основными характеристиками функционирования программ.*** длительностью исполнения, объемом базы данных и длиной текста программы. В системах реального времени временная сложность непосредственно отражается допустимым ***временем отклика*** (ожидания выходных результатов) на некоторые исходные данные. Пропускная способность комплекса программ при его исполнении на конкретной ЭВМ характеризует динамику обработки исходных данных, и сложность исполнения программ в реальном масштабе времени.

Программная сложность в первом приближении может измеряться ***количеством команд*** (строк текста), используемых в тексте данного комплекса программ. Относительная простота определения количества команд привела к широкому применению этого параметра в качестве инженерной меры программной сложности. Однако различия количества используемых байт в памяти ЭВМ на одну операцию затрудняют подсчет сложности программ по количеству команд, поэтому часто используются оценки по объему памяти, занимаемой программами (байт или слов конкретной ЭВМ). Такая оценка затрудняет сравнение программной сложности решения задач при применении ЭВМ с различной структурой команд.

В процессе эксплуатации комплекса программ основная сложность может быть сосредоточена в подготовке исходных данных и в анализе результатов. Это наиболее характерно для человеко-машинных систем обработки информации, в которых основные решения принимаются человеком. Сложность ручной подготовки, корректировки и сверки больших массивов данных (например, для материально-технического обеспечения или бухгалтерского учета) может значительно превышать программную или временную сложность функционирования программ. Для таких задач может быть велика психологическая сложность анализа и селекции результирующих данных, рассчитанных на ЭВМ.

Потребности в вычислительных ресурсах для систем в 1970-е – 80-е годы росли быстрее, чем реальные характеристики ЭВМ, что обострило проблему реализации и функционирования

сложных комплексов программ в условиях ограниченных ресурсов. Одним из важнейших и наиболее общих показателей, характеризующих возможность применения ЭВМ, являлась их производительность. Уже на стадии формулирования исследовательских, информационных или управленческих задач приходилось сопоставлять доступную производительность ЭВМ с необходимыми затратами времени для их решения или с допустимой длительностью ожидания результатов (длительностью отклика). В этом смысле использование ЭВМ являлось их использованием в реальном масштабе времени. Поэтому весьма актуальным для всех типов систем стало ***исследование эффективного распределения ограниченной производительности ЭВМ*** для решения групп различных функциональных задач и разработка методов рациональной организации вычислительного процесса в реальном времени.

Проведенные в 70-е годы исследования показали, что на эффективность использования производительности вычислительных систем ***в наибольшей степени влияли следующие факторы***:

• характеристики и параметры потоков сообщений, как от внешних абонентов, так и внутри системы, поступающих для вызова на решение определенных программ;

• характеристики программ и длительности процессов обработки сообщений различных типов и назначения;

• общая загрузка и загрузка вычислительной системы отдельными программными компонентами;

• количество процессоров и структура связей между ними, а также с общей памятью;

• методы распределения ресурсов ЭВМ, их точность и степень учета различных параметров, а также совокупная сложность и необходимая частость корректировки распределений ресурсов;

• затраты на операции распределения ресурсов в ЭВМ и их соотношение с получаемым эффектом от повышения качества распределения.

Проведенные исследования позволили ***классифицировать и оценить методы распределения вычислительных ресурсов***, а также организации вычислительного процесса в зависимости от степени неопределенности априорной информации об основных динамических характеристиках поступления внешних сообщений и их обработки:

• полностью детерминированные, использующие точную априорную информацию о временных характеристиках и потребных ресурсах времени для реализации каждой заявки на обработку информации;

• стохастические, применительно к некоторым типам или группам заявок, когда априорно известны и различимы их статистические характеристики поступления и длительности обработки;

• полностью стохастические, когда неизвестны априорные характеристики поступления и длительности обработки информации, невозможно их классифицировать и разделить на определенные типы.

При анализе распределения ресурсов, определяющими являются не столько отдельные значения интенсивностей поступления заявок и их обслуживания, сколько соотношение этих параметров, которое характеризует ***загрузку ЭВМ в реальном времени.*** Когда интенсивность поступления заявок на обработку превышает интенсивность их обслуживания, недостаток времени может приводить к неограниченному накоплению заявок в очереди на обработку. Если не принять мер по прореживанию заявок-сообщений, то они могут переполнять память ЭВМ. В то же время при малых загрузках, ЭВМ используется по производительности нерационально, и можно было бы применить для той же цели более дешевую и менее производительную ЭВМ.

Для сокращения затрат на распределения при частом их проведении ***исследованы дисциплины оперативной диспетчеризации***, которые обеспечивают приоритетные распределения, достаточно близкие к оптимальному. Эти правила основываются на предварительных экспериментальных исследованиях различных методов распределения ресурсов или на аналитических исследованиях некоторых обобщенных моделей. Анализ эффективности дисциплин диспетчеризации с учетом затрат на прерывание обработки и переключение программ показал, что при большой загрузке системы их эффективность весьма чувствительна к затратам на прерывания. При сравнительно небольших затратах на прерывания дисциплины с относительными и абсолютными приоритетами сравнимы по эффективности.

5.2. Характеристики и методы оценки качества компонентов и комплексов программ реального времени – 1975-е годы

В 1970-е годы был предпринят ряд попыток дать количественную оценку понятию ***качества программного продукта*** путем разработки ряда показателей, численно характеризующих свойства и ***требования заказчика,*** лежащие в основе этого понятия. Широко распространенному в 60-е годы подходу к программам, заключавшемуся в тезисе: «лишь бы программа работала», все более четко противопоставлялся тезис: ***«с какими характеристиками качества функционирует программа, и при каких затратах достигаются результаты».*** Стало ясным, что для превращения программирования в современную ***инженерную дисциплину,*** разработку программных продуктов следует осуществлять промышленными методами, с обязательным применением различных способов измерения характеристик качества программ, и установлением между ними причинно-следственных связей.

Для этого было необходимо научиться ***измерять и прогнозировать характеристики качества программ,*** а также изучить зависимость этих характеристик от различных параметров. Было установлено, что комплексы программ целесообразно характеризовать, прежде всего, конкретными ***функциональными показателями качества*** или показателями: назначения, номенклатуры и значениями характеристик, которые определяются целями и областью применения программных продуктов. Эти показатели весьма разнообразны, их выбор отражает основные функции конкретной программы, их номенклатуру трудно унифицировать и сопоставлять для программ, значительно отличающихся задачами. Наиболее общее качество программ – ***эффективность процессов использования*** отражает, прежде всего, назначение и выполняемые функции, вследствие чего может полностью изменяться для разных программ. Функциональные показатели в том или ином виде характеризуют эффект от использования программ с учетом затрат на их создание.

Каждый комплекс программ, кроме того, можно характеризовать ***конструктивными показателями качества,*** состав которых почти не зависит от назначения и области использования программ. Набор этих показателей весьма близок к общим характеристикам обычных сложных промышленных

систем и включает сложность, трудоемкость, конструктивную корректность, надежность, безопасность применения. Эти показатели в той или иной степени характеризуют любые программы и позволяют сопоставлять программы разного назначения по показателям качества.

Выделение конструктивных показателей из полного набора свойств, характеризующих комплексы программ, несколько условно. Особенно это относится к показателю ***корректность программ,*** которая в основном оценивается по адекватности назначения и функций программ исходным требованиям заказчика.

Вследствие роста сфер применения и ответственности функций, выполняемых программами, в 70-е годы резко возросла необходимость ***гарантирования высокого качества программных продуктов***, регламентирования и корректного формирования требований к характеристикам качества реальных программных продуктов и их достоверного определения. В результате специалисты в области теории, методов и стандартов, определяющих качество продукции, были вынуждены обратить внимание на область развития и применения понятия качества программных комплексов и систем в целом, а также на их качество, надежность и безопасность использования. Вследствие этого начало развиваться ***новое направление*** теории, методов и практики управления качеством технической продукции – ***качество программных продуктов*** [17,11].

Формализация и оценивание реального качества программных продуктов часто зависели от интуиции, вкусов и квалификации их заказчиков, пользователей и разработчиков. Имевшиеся достижения в области теории и практики управления качеством промышленной продукции, как правило, были не известны и не использовались специалистами, создающими и применяющими программы. С другой стороны, специалисты по управлению качеством продукции не исследовали и не развивали методы и культуру обеспечения и оценивания качества программных продуктов. Это во многих случаях определяло относительно низкое качество, дефекты при применении, конфликты между заказчиками и разработчиками из-за неопределенностей качества и не конкурентоспособность отечественных программных продуктов на мировом рынке.

Во многих случаях реальные контракты, спецификации требований и предварительные планы на создание программных комплексов для сложных систем подготавливались и оценивались неквалифицированно, на основе ***неформализованных представлений заказчиков и разработчиков*** о требуемых

функциях и характеристиках качества комплекса программ. Значительные системные ошибки при определении требуемых показателей качества, оценке трудоемкости, стоимости и длительности создания программного продукта являлись достаточно массовыми и типичными. Многие созданные в эти годы ***системы были не способны выполнять полностью, требуемые функциональные задачи с гарантированным качеством,*** и их приходилось долго и иногда безуспешно дорабатывать для достижения необходимого качества и надежности процессов функционирования, затрачивая дополнительно большие средства и время. Проекты оказывались неудачными или даже терпели полный провал из-за недостаточной компетентности привлекаемых разработчиков, их ***неадекватного «оптимизма»,*** а также вследствие отсутствия у них современных методов и технологий, обеспечивающих требуемое высокое качество программных продуктов.

Имевшийся опыт отдельных организаций, а также успешно реализованные ими планы и процессы разработки программных продуктов высокого качества, как правило, игнорировались, не обобщались и не использовались в качестве ***реальной базы*** для обучения специалистов, для прогнозирования и планирования новых проектов. Вследствие этого во многих проектах систем:

• не полностью реализовались цели и требования заказчика к функциям и качеству комплексов программ;

• была низка достоверность первичных оценок, необходимых бюджетов, сроков и ресурсов для разработки программ при заключении контрактов, вследствие чего не выполнялись требования заказчика к качеству;

• плохо осуществлялся контроль за ходом проектирования, из-за чего был велик риск: отсутствия у конечного продукта заданного качества; нарушения начальных планов; невыполнения функциональных и экономических требований контрактов на разработку.

В технических заданиях и реализованных проектах систематически ***умалчивались и/или недостаточно формализовались понятия и метрики качества программного продукта,*** какими характеристиками они описываются, как их следует измерять и сравнивать с требованиями, отраженными в контракте, техническом задании или спецификациях. Кроме того, некоторые из характеристик часто вообще отсутствовали в требованиях и согласованных документах, что приводило к произвольному их учету или к пропуску при испытаниях. Высокая сложность анализируемых объектов – комплексов программ, и психологическая ***самоуверенность ряда специалистов в***

собственной «непогрешимости», часто приводили к тому, что реальные характеристики качества функционирования программ оставались неизвестными не только для заказчиков и пользователей, но также для самих разработчиков.

Основные проблемы этого нового направления в области качества, по существу, были близки к типичным проблемам исследования и управления качеством других видов сложных технических систем, и в конце 70-х годов были поставлены в НИР ПРОМЕТЕЙ. Задачи по анализу и исследованию качества программных комплексов включали:

• разработка понятий и систематизация достаточно полного набора регламентированных базовых характеристик и атрибутов качества программных продуктов, адекватно отражающих специфику этого вида продукции в различных областях использования;

• создание методов выбора и адаптации комплектов базовых характеристик качества для формирования сопоставимых требований к качеству конкретных проектов с учетом особенностей, назначения и сферы их применения;

• исследование и разработка методов, процессов и технологий реализации заданных требований к характеристикам качества программных продуктов в условиях ограниченных ресурсов на поддержку жизненного цикла комплексов программ;

• исследование и разработка методов испытаний программных продуктов и оценивания достигнутых характеристик качества на различных этапах их жизненного цикла;

• подготовка методов и специалистов, способных обеспечивать и контролировать высокое качество жизненного цикла крупных комплексов программ для различных сфер применения.

Исследования показали, что ***радикальное повышение качества отечественных программных продуктов и обеспечение их конкурентоспособности*** возможно только на базе внедрения регламентированных технологий и систем качества, поддерживающих и контролирующих весь жизненный цикл производства программного продукта. ***Основой обеспечения высокого качества сложных комплексов программ*** является ***полнота тестирования программных компонентов – модулей*** (ПМ) [19]. Для отладки программ наиболее полно в 80-е годы исследованы методы и характеристики тестирования структуры ПМ и обработки в них потоков данных (Борис Аронович Позин). Тестирование структуры ПМ позволяло выявлять наиболее грубые и опасные ошибки в программах, которые могли приводить к

резким отличиям результатов их исполнения, от предполагаемых эталонов. Имевшиеся всегда реальные ограничения ресурсов на разработку программных компонентов, вызывали необходимость рационального их использования, и систематизированного применения различных методов в целях достижения наивысшего качества программ.

Была исследована сложность тестирования типовых структур модулей при различных критериях выделения маршрутов, а также достигаемая их корректность в зависимости от объема тестов. Для логических программ наибольшее значение имело тестирование при значениях исходных данных, определяющих реализуемые маршруты исполнения программ. Систематизированы и оценены по эффективности методы тестирования потоков данных. Для вычислительных программ особенно важны были методы тестирования корректности обработки каждой переменной и оценки точности результатов вычислений.

Детерминированное тестирование структуры программных модулей имело целью проверять корректность выделенных маршрутов исполнения программ и обнаружение в основном логических ошибок формирования маршрутов. На практике при отсутствии упорядоченного анализа потоков управления некоторые маршруты в программе могли оказаться пропущенными при тестировании, поэтому первая задача, которая решалась при исследовании тестирования структуры программ, – это ***получение информации о полной совокупности реальных маршрутов исполнения в каждой программе.*** Такое представление маршрутов позволяло упорядоченно контролировать достигнутую степень проверки маршрутов и в некоторой степени предохраняло от случайного пропуска отдельных не трестировавшихся. В результате значительно повышалось достигаемое качество программ, и тестирование приобретало планируемый систематический характер.

Сложность и трудоемкость тестирования потоков управления определяются комбинаторным характером схем принятия решений во многих программных модулях, что приводит к большому числу маршрутов обработки информации в программах. Анализ критериев тестирования и выделение тестируемых маршрутов удобно было проводить, используя графовые модели программ. При планировании тестирования структуры программ были исследованы, прежде всего, ***две задачи,*** формирование критериев выделения маршрутов для тестирования и выбор стратегий упорядочения выделенных маршрутов.

В ряде случаев была подтверждена ***адекватность использования структурной сложности программ для оценки трудоемкости тестирования,*** вероятности не выявленных ошибок и затрат на разработку программных модулей в целом. В качестве исходной информации при тестировании структуры программ использовалась схема связей между операторами текста программы. По этой схеме выделялся полный набор маршрутов исполнения ПМ, подлежащих отладке. Для каждого выделенного маршрута по тексту программы формировался набор условий, определяющих его реализацию при соответствующем тесте. ***Сложность тестирования программных модулей*** можно было оценивать по числу маршрутов, необходимых для их проверки, или более полно – по суммарному числу условий, которое необходимо задать в тестах для прохождения всех маршрутов программы.

Показано, что маршруты исполнения ПМ можно было разделить на ***два вила:*** маршруты исполнения преимущественно вычислительной части программы и преобразования непрерывных переменных; маршруты принятия логических решений и преобразования логических переменных. Маршруты первого вида обычно логически проще и короче, чем второго, и предназначены для преобразования величин, являющихся квантованными результатами измерения некоторых непрерывных физических характеристик (непрерывные переменные). Такие переменные связаны условиями гладкости, т. е. условиями малых изменений производных этих переменных по времени или по другим параметрам.

При ***оценке сложности вычислительных маршрутов программ*** необходим был учет числа операндов, участвующих в вычислениях. Кроме того, исходные и результирующие данные при тестировании должны были принимать несколько значений. Во всем диапазоне исходных переменных следовало выбирать характерные точки (предельные и промежуточные значения), при которых проверяется программа. В особых точках значений и сочетаний переменных и в точках разрыва функции необходимо было планировать дополнительные проверки.

Проведенные ***исследования позволили оценивать достигаемую полноту тестирования*** при заданном числе тестов. Влияние критериев выбора маршрутов, стратегий их упорядочения и вероятностей ветвления в вершинах на вероятность проявления ошибок при тестировании программ более наглядно отражается на графах программ. Для выбора маршрутов тестирования наиболее подробно были исследованы критерии, при которых каждая дуга графа программы

проверяется хотя бы один раз и маршруты различаются хотя бы одной дугой.

Ветвления в программах реального времени специализированных ЭВМ происходили через 5 – 10 операторов текста программ, поэтому число маршрутов исполнения ациклических ПМ было пропорционально их размеру, выраженному числом строк текста программ. Реализация каждого маршрута ПМ определяется числом условий, которые необходимо задать для его исполнения при тестировании. Поэтому полное ***число условий в тестях для покрытия тестями структуры ПМ было пропорционально квадрату строк текста программы*** в модуле и соответственно быстро возрастало при увеличении размера ПМ. На этой основе было показано, что при разработке ПМ целесообразно учитывать рациональное ***ограничение размеров модулей на уровне трехсот строк текста,*** что соответствует приблизительно тридцати альтернативам в ациклических программах. Поэтому при разработке ПМ был ***рекомендован рациональный размер программ модулей в пределах 100–200 строк текста*** на автокоде, для полного тестирования которых достаточно использовать 10–20 тестов с суммарным числом условий до 100. При превышении рекомендуемых размеров ПМ их трудно протестировать полностью, и целесообразно было делить на более мелкие компоненты, доступные для полного покрытия тестами.

Исследование структуры реальных программ, заказных специализированных ЭВМ позволило выявить около трети ПМ, ***содержащих циклы.*** При наличии в ПМ циклов, число необходимых тестов быстро возрастало в зависимости от числа маршрутов в теле цикла и числа различных итераций цикла, необходимых для проверки этой части ПМ. Рекомендовано при планировании тестирования ПМ по возможности предварительно выделять и отдельно тестировать циклы, а затем их включать в ациклическую часть программ для полного тестирования ПМ.

Продолжительность разработки программ всегда ограничена, что обычно ***не позволяет обеспечить полное покрытие тестами структуры ПМ.*** Поэтому при тестировании целесообразно упорядочивать маршруты исполнения программы и начинать тестирование либо с маршрутов, самых длинных по числу строк текста программы, либо с наиболее сложных по числу ветвлений в графе программы, либо с наиболее вероятных при исполнении ПМ. Было показано, что при одинаковом числе вершин ветвления в программах, сложность тестов, обеспечивающих достаточно низкую вероятность ошибок, в зависимости от структуры программы и критерия выделения маршрутов, может различаться

почти на два порядка. Графы реальных программ практически всегда являются несимметричными как по структуре, так и по вероятностям исполнения маршрутов в ПМ. Это свойство целесообразно использовать при упорядочении последовательности, выделяемых при планировании для тестирования маршрутов, начиная с наиболее вероятных. Выполненные исследования позволили выработать рекомендации, как ***рационально планировать тестирование, в целях получения максимальной корректности*** модулей при ограниченных ресурсах на отладку. Для этого были созданы ***инструментальные средства автоматизированного планирования отладки программных компонентов.***

5.3. Методы оценивания и обеспечения корректности и надежности программных продуктов – 1975-е годы

В конце 1970-х годов было установлено ***отсутствие тождественности*** между понятиями ***правильной и надежной программы реального времени*** [17]. Понятие правильной, или корректной, программы может рассматриваться ***статически,*** вне временного функционирования. Корректная программа должна была обеспечивать выходные данные, соответствующие эталонным требованиям, в области изменения исходных данных заданных техническим заданием или спецификацией. Степень правильности программы можно характеризовать вероятностью попадания в область исходных данных, которая предусматривалась требованиями спецификации, но не была проверена при тестировании и испытаниях. В этой области исходных данных не гарантируются эталонные результаты из-за возможных ошибок в программе, не обнаруженных при отладке. При этом корректность программы не определена вне области данных, заданной спецификацией, и не зависит от динамики функционирования программ в реальном времени. Таким образом, ***некорректность исполнения программы*** определяется совмещением событий: попаданием исходных данных в область, заданную требованиями спецификации, но не проверенную при тестировании и испытаниях и наличием ошибки в программе при обработке таких данных.

Надежная программа, прежде всего, должна обеспечивать низкую вероятность отказа в процессе ее реального функционирования. Быстрое реагирование на искажения программ, данных или вычислительного процесса и восстановление работоспособности за время, меньшее, чем порог между сбоем и отказом, позволяют обеспечивать высокую надежность программ. При этом некорректная программа может функционировать в принципе абсолютно надежно. Действительно, если каждое появление реальных исходных данных, попадающих в области, не проверенные при тестировании и стимулирующих неправильные результаты, не приводит к событиям, соответствующим отказу, то такая программа функционирует ***безотказно и надежно, хотя и не всегда правильно.***

Совершенно корректная программа определена в области исходных данных, заданных требованиями технического задания.

Однако в реальных условиях за счет различных причин исходные данные могут попадать в область, не проверенную при отладке и не соответствующую требованиям спецификации. При таких исходных данных правильная программа не проверялась и может потерять работоспособность за счет конфликта между исходными данными и программой. В результате формально правильная программа окажется ненадежной, прежде всего, из-за системных ошибок при задании области изменения исходных данных. Если при тех же исходных данных восстановление происходит за время, меньшее, чем пороговое время между отказом и сбоем, то такие события не влияют на основной показатель надежности – ***наработку на отказ.*** Следовательно, отказ при функционировании программы является понятием ***динамическим.***

Таким образом, показатели надежности программ должны учитывать ***характеристики восстановления после возникновения отказовой ситуации в*** процессе функционирования. Кроме того, корректная и надежная программы различаются областями изменения исходных данных, которые определяют степень некорректности или степень ненадежности.

В некоторой области изменения исходных данных, корректность и надежность программы оказываются коррелированными, что соответствует данным, определенным техническим заданием и не проверенным при тестировании. Однако и при таких исходных данных программа может иметь высокие показатели надежности, если восстановление производится оперативно в пределах интервала времени, ограничивающего события сбоев.

Так как в программах нет необходимости «ремонта» компонентов с участием человека, то нужно добиваться ***высокой автоматизации программного восстановления функционирования.*** Автоматизируя процесс и сокращая время восстановления, можно преобразовать отказы в сбои и тем самым улучшить показатели надежности функционирования системы в реальном времени. Для решения этой задачи в комплексе программ ***должны быть средства, позволяющие*** [17]:

• проводить систематический контроль и оперативно обнаруживать аномалии процесса функционирования или состояния программ и данных;

• диагностировать обнаруженные искажения;

• вырабатывать решения и выбирать методы и средства оперативного восстановления;

• реализовывать оперативное восстановление нормальной

работоспособности;

• регистрировать каждый происшедший сбой или отказ и обобщать с данными предыдущих искажений для выявления систематических случаев, требующих доработки программ или аппаратуры.

Высокое качество программного продукта достигается при определенных затратах на их разработку и отладку. Значительную долю в них составляют затраты на тестирование и испытания комплекса программ, особенно в системах реального времени. Исследование факторов, влияющих на затраты ресурсов при создании программ, позволило рационализировать их использование и добиваться заданного качества при минимальных или допустимых затратах. Повышению качества функционирования комплекса программ способствует ***введение избыточности в программы и данные.*** Особенно большое влияние может оказывать избыточность на надежность решения задач в реальном времени. При этом возможно снижение затрат на отладку и частичное обеспечение необходимой надежности программ за счет средств повышения помехоустойчивости, оперативного контроля и автоматического восстановления функционирования программ.

В программных продуктах реального времени, ***для обеспечения высокой надежности*** их функционирования в 80-х годах было предложено ***максимально быстро обнаруживать искажения,*** возможно, точно классифицировать тип уже имеющихся и возможных последствий дефектов, а также проводить мероприятия, обеспечивающие ***быстрое восстановление нормального функционирования программного продукта.*** Неизбежность ошибок в сложных комплексах программ, искажений исходных данных и аппаратных сбоев привели к необходимости регулярной автоматической проверки процесса исполнения программ и сохранности данных. В процессе проектирования недостаточно было создавать корректные программы, выдающие верные результаты при идеальных исходных данных и абсолютном отсутствии любых возмущений. Требовалось разрабатывать ***надежные и безопасные программы, устойчивые к различным, негативным возмущениям*** и способные сохранять достаточное качество результатов в реальных условиях функционирования.

При этом не обязательно сразу устанавливать причины искажений, главная ***задача сводится к максимально быстрому восстановлению нормального функционирования и ограничению последствий проявления дефектов.*** Временная и программная избыточность комплекса программ состоит в

использовании некоторой части производительности и памяти ЭВМ для оперативного контроля исполнения программ и восстановления ***(рестарта***) вычислительного процесса. Величина избыточности зависит от требований к надежности и безопасности функционирования систем обработки информации и обычно должна находиться в пределах от 5 – 10 % ресурсов однопроцессорной ЭВМ или до трех, четырехкратного дублирования отдельной машины в мажоритарных вычислительных комплексах.

В 1970-е – 80-е годы дефекты и ошибки в сложных комплексах программ стратегических ракет неоднократно приводили к их гибели с огромным ущербом. Резко обострилась проблема ***оценивания и повышения безопасности применения мобильных и бортовых программных продуктов.*** Единичные и непредсказуемые ситуации и их катастрофические последствия, требовали практически полного исключения, и отличали от дефектов в аппаратуре, надежность и безопасность которой можно рассчитывать аналитически. Необходимо было создавать систему защитных барьеров и самоконтроля исполнения программ для обеспечения гарантированной работоспособности и ***безопасности применения сложных систем управления*** на базе программных продуктов.

Функциональная безопасность программных продуктов должна гарантировать корректное управление динамическими объектами и системами, используемыми в авиации, космосе, на транспорте, которые создаются с применением технологий и инструментальных средств, по существу, аналогичных применяемым для разработки других классов сложных комплексов программ реального времени. Любая ***стратегия функциональной безопасности*** должна учитывать не только все элементы в каждой конкретной системе (например, датчики, управляющие устройства и исполнительные механизмы), но также все системы и программные компоненты, создавать суммарную комбинацию систем, обеспечивающих безопасность. Прямое измерение достигнутой функциональной безопасности объектов и систем сталкивается с большими трудностями, прежде всего вследствие необходимости фиксировать редкие, непредсказуемые отказовые ситуации с неожиданным ущербом. Соответственно усилия разработчиков систем и программных средств сосредоточивается на методах регламентирования и обеспечения качества жизненного цикла таких объектов и систем. Поэтому основное содержание документов в области функциональной безопасности составляют ***методы, процессы и технологии обеспечения высокой безопасности производства***

систем и почти не уделяют внимание ***определению значений*** достигаемой безопасности.

Планирование функциональной безопасности программных продуктов должно определять стратегию получения, разработки, компоновки, верификации, аттестации и модификации комплекса программ в той мере, которая необходима для требуемого уровня соответствия комплексу требований по безопасности. Цикл обеспечения безопасности при разработке программного продукта должен быть ***выбран и задан при планировании безопасности,*** приспособлен для практических нужд конкретного проекта или предприятия. Каждый этап цикла обеспечения безопасности должен быть подразделен на элементарные операции в рамках сферы действия, а также входных и выходных данных, оговоренных для каждого этапа производства. Полный перечень этапов цикла обеспечения безопасности ориентирован для больших вновь разрабатываемых систем. В малых системах может оказаться целесообразным объединять этапы проектирования программ и архитектурного проектирования объекта или системы. Спецификация требований по безопасности ***должна быть достаточно подробной*** для того, чтобы конструкция и ее выполнение достигали требуемого соответствия комплексу требований по безопасности, и можно было произвести оценку достигнутой функциональной безопасности.

Должно быть произведено планирование с целью определения шагов, как процедурных, так и технических, которые должны быть использованы для демонстрации того, что программный комплекс ***удовлетворяет требованиям по его безопасности.*** План ***аттестации безопасности*** должен содержать: подробные указания о том, когда должна производиться аттестация; кто должен производить аттестацию; определение ответственных режимов эксплуатации технических средств; техническую стратегию аттестации; требуемые условия окружающей среды, в которой должна производиться аттестация; критерии ее прохождения; политику и процедуры оценки результатов аттестации.

Целью требований является обеспечение соответствия объединенной системы, заданным требованиям по безопасности программного продукта ***при заданном уровне безопасности внешней среды и системы.*** Для каждой функции безопасности, в документации по аттестации должны быть указаны: использованный вариант плана аттестации; функция, проходившая аттестацию (путем анализа, экспертизы или экспериментальных испытаний) вместе со ссылками на план аттестации; инструментальные средства и оборудование, данные

об их калибровке; результаты аттестации; расхождение между ожидаемыми и полученными результатами. Если обнаружены расхождения между ожидаемыми и полученными результатами, следует принять решение о том, продолжать ли аттестацию или сделать заявку на изменения и вернуться к более ранней документация, являющейся частью требований и результатов аттестации программного продукта. Программный комплекс должен быть проверен тестированием при имитации:

• входных сигналов, существующих при нормальной эксплуатации;

• ожидаемых происшествий и аномалий функционирования;

• нежелательных ситуаций, требующих вмешательства системы контроля и корректировки безопасности.

К результатам аттестации предъявляются следующие требования: ***испытания должны показать,*** что все заданные требования по безопасности программного продукта выполняются правильно, и что продукт не выполняет не предназначенных ему функций. По каждому испытанию и его результатам должна быть составлена документация для проведения последующего анализа и независимой оценки соответствия с требуемым уровнем безопасности. Выбор методов оценки не гарантирует сам по себе, что будет достигнуто соответствие комплексу требований по безопасности. Нарушения безопасности комплексов программ возникают вследствие преднамеренного использования или случайной активизации ***уязвимостей*** при применении системы по назначению. Уязвимости могут возникать ***вследствие недостатков:***

• требований – продукт или система могут обладать требуемыми от них функциями и свойствами, но содержать уязвимости, которые делают их непригодными или неэффективными в части безопасности применения;

• проектирования – продукт или система не отвечают спецификации, и/или уязвимости, являются следствием некачественного проектирования или неправильных проектных решений;

• эксплуатации – продукт или система разработаны в полном соответствии с корректными спецификациями, но уязвимости возникают как результат неадекватного управления при эксплуатации.

Оценка, и утверждение целей функциональной безопасности требуется для демонстрации заказчику или пользователю, что установленные цели проекта адекватны проблеме его безопасности. Следует сопоставлять цели безопасности с ***идентифицированными угрозами,*** которым они

противостоят, и/или с политикой и предположениями, которым они должны соответствовать. Некоторые могут зависеть от требований безопасности системы, выполняемых ее средой. В этом случае требования безопасности, относящиеся к внешней среде, необходимо оценивать в контексте требований к системе.

Руководства по безопасности определяют требования, направленные на обеспечение понятности, достаточности и законченности эксплуатационной документации, представляемой разработчиком. Эта документация, которая содержит две категории информации (для пользователей и администраторов), является важным фактором безопасной эксплуатации объекта и программного продукта. Требования к руководству пользователя должны обеспечивать возможность эксплуатировать объект безопасным способом. Руководство – основной документ, имеющийся в распоряжении разработчика, для предоставления пользователям необходимой общей и специфической информации о том, как правильно использовать функции безопасности.

Целесообразно идентифицировать компоненты или части их, критические с точки зрения безопасности, сбой в которых может привести к отказовой ситуации. Если имеется такой компонент, следует предусмотреть стратегию обеспечения его защиты. Стратегия должна гарантировать методы, при которых требования, проект, реализация и эксплуатационные процедуры, минимизируют или устраняют потенциальные нарушения безопасности программного продукта. Следует проанализировать требования контракта, относящиеся к использованию ресурсов аппаратных средств ЭВМ (например, максимально возможная производительность процессора, объем памяти, пропускная способность устройств ввода /вывода).

Разработчик должен принимать участие в определении и документировании проектных решений системного уровня. Он ответствен за выполнение всех требований и демонстрацию этого выполнения посредством ***квалификационного тестирования.*** Реализация проектных решений, действующих как внутренние требования, должна быть подтверждена внутренним тестированием разработчика.

Следует осуществлять контроль за критическими для выполнения контракта ситуациями, которые могут возникнуть во время разработки комплекса программ. Необходимо выявлять, идентифицировать и анализировать потенциальные технические, стоимостные или временные критические ситуации и риски; разработать стратегии для предотвращения или устранения таких ситуаций; регистрировать возможные риски и стратегии их

предотвращения и реализовать эти стратегии в соответствии с Планом.

Необходимо регламентировать обеспечение безопасности в ***должностных инструкциях*** и при выделении ресурсов, а также обучение специалистов правилам контроля безопасности, реагированию на события, таящие угрозу безопасности и уведомлению об отказах программного продукта. Должны быть разделены физическая безопасность системы и безопасность окружающей среды, программных средств разработки и рабочих программ.

5.4. История формирование экономики программной инженерии в 1980-е годы

В 1950-е – 60-е годы к созданию программ для ЭВМ исторически подходили, как к «искусству и художественному творчеству» отдельных специалистов или небольших групп. При этом считалось, что ***невозможно применять, какие-либо экономические характеристики*** для определения стоимости и результатов такого творчества, и они оценивались только с позиции качества выполняемых функций и ***«эстетики»*** их реализации. Такие программы создавались преимущественно для получения конкретных результатов исследований или для анализа относительно простых процессов обработки информации. Они не предназначены для массового тиражирования и распространения как программный продукт на рынке, их оценивали качественно и интуитивно, преимущественно как «художественные произведения». При этом, как правило, не было конкретного заказчика-потребителя, определяющего требования к программам и их финансирование, программы не ограничивали допустимой стоимостью, трудоемкостью и сроками их создания, требованиями обеспечения заданного качества и документирования. Их разработчики не применяли регламентирующих, нормативных документов, вследствие чего жизненный цикл таких изделий имел ***не предсказуемый характер*** по качеству и стоимости основных результатов «творчества».

Для небольших относительно простых программных комплексов, во многих случаях, достаточно достоверными могли быть ***интуитивные оценки требуемых экономических ресурсов,*** выполняемые опытными руководителями, реализовавшими несколько аналогичных проектов. Такие оценки ***отличались большими ошибками при планировании экономических характеристик —*** сроков, трудоемкости и стоимости создания сложных программных продуктов. В большинстве случаев, это приводило к значительному запаздыванию завершения разработок и превышению предполагавшихся затрат ресурсов. Вследствие пренебрежения тщательным экономическим обоснованием, до 15 % проектов сложных программных комплексов не доходило до завершения, а почти половина проектов не укладывалось в выделенные ресурсы, бюджет и сроки, не обеспечивало требуемые характеристики качества. Отставание сроков внедрения некоторых больших промышленных

и оборонных систем управления и обработки информации полностью ***зависело от неготовности для них программных продуктов.***

Массовое создание сложных и дорогих программных продуктов промышленными методами и большими коллективами специалистов (в основном для оборонных систем) вызвало необходимость их достоверного ***экономического прогнозирования и анализа,*** четкой организации производства, планирования работ по затратам, этапам и срокам реализации. Для решения этих задач в 70-е годы начала формироваться новая область знания и инженерная дисциплина – ***экономика, создания сложных программных продуктов*** [15, 11]. Необходимо было освоить анализ и оценивание конкретных факторов, влияющих на экономические характеристики проектов программных продуктов вследствие реально существующих и потенциально возможных воздействий и ограничений ресурсов проектов комплексов программ. Это привело к появлению ***новой области экономической науки и практики – экономики проектирования, производства и жизненного цикла сложных программных продуктов,*** как части экономики в промышленности и вычислительной технике в составе общей экономики некоторых предприятий. Ее основной задачей являлись ***анализ, прогнозирование, эффективное управление, распределение ресурсов и экономное их использование.***

Развитие этой области экономики было ***связано с большими профессиональными и психологическими трудностями,*** типичными для новых разделов современной науки и техники, появляющимися на стыке различных областей методов и знаний. В данном случае особенности состояли в том, что ***менеджеры и разработчики*** сложных комплексов программ, как правило, не знали даже основ экономики промышленного производства сложной технической продукции, а ***экономисты*** не представляли сущность и свойства объектов разработки – программных продуктов, а также особенностей технологических процессов их производства и применения. Широкий спектр технических характеристик таких объектов, количественных и качественных факторов, которые с различных сторон характеризуют содержание компонентов и комплексов программ, и невысокая достоверность оценки их значений, определяли значительные трудности при попытке описать и измерить свойства и качество создаваемых или используемых сложных программных продуктов ***для их экономической оценки и прогнозирования характеристик.***

Приступая к созданию сложных технических проектов,

заказчики и исполнители, прежде всего, должны были ***понять целесообразна ли их разработка*** и оценить возможную эффективность применения готового продукта, оправдаются ли затраты на его производство и использование. Поэтому такие технические проекты в промышленности традиционно начинались с ***анализа и экономического обоснования*** предстоящего производства и применения предполагаемого продукта или системы. Заказчику и возможным потребителям результатов проекта необходимо оценивать реальную потребность в его создании и конкурентоспособность, а потенциальному разработчику предполагаемого продукта, проводить оценку реализуемости проекта при условиях и ресурсах, предлагаемых заказчиком. Однако часто разработчики не в состоянии привести заказчику или руководителю проекта достаточно обоснованные ***доказательства не реальности выполнения выдвигаемых требований*** к продукту при предложенных ограниченных значениях бюджета и сроков. Руководители конкретных проектов обычно не в состоянии достаточно обоснованно определять, сколько времени и затрат труда потребуется на каждый этап проекта системы, и не могут оценить, насколько успешно выполняется план производства. Это, как правило, означает, что проект системы с самого начала ***выходит из-под экономического контроля*** и возможна катастрофа с реализацией и завершением проекта всей системы в требуемый срок с заданным бюджетом и качеством.

Создание в 1970-е – 80-е годы сложных программных комплексов как ***производственной продукции*** существенно повысило ***актуальность экономического обоснования, необходимость прогнозирования и измерения*** процессов производства и их характеристик качества. Основной целью производства многих программных продуктов является повышение эффективности промышленных систем обработки информации и/или управления объектами и административными системами, в которых применяются сложные комплексы программ. Такими системами могли быть средства автоматизированного управления в авиации, космическими системами, энергосистемами, системами административного управления, автоматизации проектирования.

В ряде случаев программные продукты невозможно или очень трудно характеризовать ***непосредственной экономической эффективностью применения*** (например, оборонных систем) в зависимости от затрат ресурсов и целесообразно из анализа исключать характеристики эффективности программных продуктов и сопутствующие ей функциональные критерии

качества. При планировании производства сложных программных продуктов часто имеется определенный ***заказчик-потребитель, который способен задать*** основные цели, функции, требуемые характеристики качества и обеспечить ресурсы и бюджет для реализации проекта. Однако иногда инициатором разработки комплекса программ является разработчик-поставщик, который самостоятельно принимает все решения о его производстве за счет собственных ресурсов и предполагает возместить затраты путем реализации программного продукта на рынке. Таким образом, ***при экономическом анализе и обосновании проектов сложных комплексов программ возможны два сценария.***

• создание и весь жизненный цикл комплексов программ ориентируется на массовое тиражирование и распространение их на рынке, среди заранее не известных пользователей, в различных сферах и внешней среде применения; при этом отсутствует конкретный внешний потребитель – заказчик, который определяет и диктует основные требования к программному продукту, выделяет ресурсы и финансирует проект;

• производство программного продукта предполагается с определенным, относительно небольшим тиражом, с известной областью и внешней средой применения для конкретного потребителя – заказчика, который определяет требования к функциям, характеристикам качества, финансирует и выделяет ресурсы.

Эти сценарии ***принципиально различаются методами экономического анализа*** и обоснования их экономических характеристик. ***Первый сценарий*** базируется на маркетинговых исследованиях рынка программных продуктов и на стремлении поставщика занять на рынке достаточно выгодное место. Для этого ему необходимо определить наличие на рынке всей гаммы близких по назначению и функциям продуктов, оценить их эффективность, стоимость и применяемость, а также ***возможную конкурентоспособность*** предполагаемого к разработке программного продукта для потенциальных пользователей и их возможное число. Следует оценивать рентабельность затрат на создание нового продукта, выявлять факторы, функциональные, экономические и конструктивные характеристики качества, которые способны привлечь достаточное число покупателей и оправдать затраты на предстоящую разработку. При выборе продукции покупатель обычно стремится максимизировать это соотношение как за счет поиска продуктов с наибольшими эффективностью и качеством, так и за счет минимальной стоимости покупаемого продукта. В этом сценарии при организации производства вся ответственность за цели, функции

и показатели качества продукта ложатся на его руководителей.

Второй сценарий предполагает наличие определенного заказчика – потребителя конкретного программного продукта, который должен соответствовать его ***формализованным и утвержденным техническим и экономическим требованиям.*** Он выбирает конкурентоспособного поставщика – производителя продукта, которого оценивает на возможность реализовать проект с необходимым качеством с учетом ограничения требуемых бюджета, сроков и других ресурсов. При этом результаты разработки не обязательно подлежат широкому тиражированию, могут не поступать на рынок, маркетинговые исследования для таких проектов являются второстепенными и предварительно могут не проводиться.

Однако для заказчика и разработчиков при заключении контракта необходимо достаточно достоверное ***прогнозирование требований*** к программному продукту и ***экономическое обоснование необходимых ресурсов*** по трудоемкости, стоимости, срокам реализации и другим показателям. Заказчик заинтересован в получении программного продукта высокого качества при минимальных затратах, а разработчик желает получить максимальную оплату за созданный продукт и достаточные ресурсы на его производство. Противоположность интересов поставщика и потребителя при оценке экономических характеристик, стоимости и других ресурсов проекта, требует ***поиска компромисса,*** при котором производитель программного продукта не продешевит, а заказчик не переплатит за конкретные выполненные работы и весь проект.

Поэтому оба партнера ***заинтересованы в достоверном экономическом прогнозировании и*** обосновании проектирования и производства программного продукта. Для этого должен быть подготовлен согласованный между заказчиком и разработчиком ***договор,*** в котором определены ***цели и задачи проекта, требуемые характеристики продукта, доступные экономические и другие ресурсы для его реализации*** [11, 15]. Эти данные должны быть предварительно сбалансированы, и обеспечивать реализацию целей проекта при выделенных ресурсах с минимальным допустимым риском.

Методы и достоверность ***экономического анализа производства*** сложных программных продуктов можно разделить на ***две части,*** существенно различающиеся особенностями производственных процессов, экономическими характеристиками и влияющими на них факторами. ***В первой части жизненного цикла.*** производятся: системный анализ, проектирование, разработка, тестирование и испытания первой (базовой) версии

программного продукта. Номенклатура работ, их трудоемкость, длительность и другие характеристики на этих этапах производства существенно зависят от ***свойств и характеристик создаваемого продукта,*** требуемых показателей качества, внешней и технологической среды разработки. При этом первостепенное значение имеют размеры, архитектура, сложность функций, количество, состав и взаимосвязи компонентов комплекса программ, которые являются базой для оценивания экономических характеристик производства конкретного продукта. Изучение подобных зависимостей для различных прототипов программных продуктов позволяет достаточно достоверно прогнозировать состав и основные экономические характеристики производства, планы и графики работ для вновь создаваемых продуктов.

Вторая часть жизненного цикла, отражающая применение, сопровождение и модификацию версий программного продукта, связана не только с экономическими характеристиками продукта и среды производства, они зависят ***от величин и интенсивности изменений*** версий, сложности и стоимости каждой модификации программного продукта. После того как программный продукт создан и испытан, в ряде случаев он становится недоступным для разработчиков и применяется неизменным до внедрения очередной версии после модернизации системы. В ***процессе сопровождения*** программы могут подвергаться эпизодическим корректировкам, которые должны регистрироваться, накапливаться и передаваться пользователям экземпляров программного продукта. Необходимо обеспечивать пользователям адекватность технологической документации каждой версии эксплуатируемого продукта в любой момент времени.

В 80-е годы выделились и сформировались ***три группы наиболее актуальных экономических зада ч*** для их эффективного внедрения в ***практику программной инженерии,*** анализ экономики программных продуктов; создание и организация экономически эффективного их проектирования и производства; подготовка и обучение специалистов для экономически эффективного проектирования и производства программных продуктов.

Анализ экономики проектирования и производства программных продуктов включал:

исследование методов оценивания и характеристики экономической деятельности предприятий при проектировании и производстве сложных программных продуктов (прибыль, доход, рентабельность, производительность труда) в различных отраслях промышленности;

• определение, анализ и классификация экономических характеристик – трудоемкости, стоимости, длительности производства, и реальных факторов, влияющих на экономику проектирования и производства программных продуктов;

• измерение, сбор и обобщение реальных экономических характеристик процессов проектирования и производства сложных программных продуктов на предприятиях и в отраслях промышленности;

• анализ экономики, планирования и достоверности прогнозирования экономических характеристик реальных проектов программных продуктов;

• разработка методов оценивания и выбора, квалифицированных коллективов специалистов-подрядчиков, способных создавать сложные программные продукты и базы данных требуемого качества в разумные сроки с учетом ограничений на используемые ресурсы.

Создание и организация экономически эффективного проектирования и производства программных продуктов включало:

создание эффективной технологии и экономики деятельности предприятия, обеспечивающей рациональное сочетания целей, стратегий и использования доступных ресурсов, необходимых для достижения целей производства программного продукта высокого качества;

• создание эффективных методов экономического и технического формирования требований к функциям, характеристикам и качеству программного продукта, с учетом экономических и ресурсных ограничений проектирования и производства на предприятии;

• создание методов экономического обоснования и прогнозирования разработки комплексной системы качества, планов и Программы испытаний, методологии и инструментальных средств, обеспечивающих требуемые качество, надежность и безопасность функционирования программного продукта;

• обеспечение экономических характеристик производства, предсказуемого и управляемого уровня достигаемого качества программного продукта, непосредственно зависящего от инструментария производства, систем качества и эффективности технологий, используемых на этапах жизненного цикла;

• обеспечение и удостоверение требуемых функций и качества готового программного продукта в процессе тестирования и испытаний с полным комплектом адекватной эксплуатационной и технологической документации.

Подготовка и обучение специалистов для экономически эффективного проектирования и производства программных продуктов составляли:

• обучение и подготовка руководителей и специалистов по экономике производства сложных программных продуктов, освоение современных методов экономического анализа, оценивания и прогнозирования процессов и необходимых ресурсов при производстве комплексов программ;

• обеспечения экономически эффективного производства программных продуктов, подготовка и обучение ***«команд»*** системных аналитиков, архитекторов и менеджеров проектов, специалистов по комплексированию, верификации, тестированию и сертификации сложных программных продуктов, по испытаниям и обеспечению их качества;

• выделение и обучение специалистов, ответственных за анализ, оценивание и прогнозирование экономических характеристик производства, за соблюдение экономически эффективной промышленной технологии создания и совершенствования программных продуктов;

• освоение руководителями и специалистами современных экономически эффективных технологических задач, методов и стандартов промышленного создания и развития сложных, тиражируемых программных комплексов требуемого высокого качества;

• руководители и специалисты для обеспечения высоких экономических характеристик при производстве сложных комплексов программ должны были освоить типовые, стандартные решения и использование апробированных заготовок программных компонентов, не требующих при их применении высококвалифицированного творческого труда.

Для развития ***экономики производства программных продуктов*** требовалось обобщение и освоение накопленного опыта и опубликованных данных в этой новой области экономики. Для квалифицированного прогнозирования и отслеживания экономики на различных этапах производства сложных комплексов программ, ***необходимо обучение специалистов новой экономической профессии*** по специальной программе в составе общей квалификации ***«программная инженерия».*** Такие специалисты – экономисты должны были входить в состав проектных коллективов или предприятий, создающих сложные программные продукты.

Для ряда систем управления производство комплекса программ требовало 2–3 года (и более), что близко к длительности создания аппаратной части систем.

Такая продолжительность проектирования была приемлемой при создании широкого класса комплексов программ, пока не удавалось заметно сократить продолжительность создания аппаратной части систем. Специфика программного продукта, существенно отличающая его от вычислительной техники, состояла в низкой относительной стоимости массового выпуска копий программ при серийном производстве. Если для большинства случаев стоимость серийных изделий аппаратуры снижалась по сравнению с опытным образцом в 3 – 10 раз, то стоимость разработки сложных комплексов управляющих программ могла превышать в 100 – 1000 раз затраты на их копирование, установку и проверку в каждой серийной системе. Поэтому необходимо было такое ***структурное построение комплекса программ,*** которое позволяло бы легко проводить частичную замену программ, настройку на различные характеристики внешней среды, а также переносить некоторые отработанные программы в системы другого типа и назначения.

Этап отечественной ***истории программной инженерии*** в 80-е годы значительно отличался от остальных этапов высокой интенсивностью разработок и активным внедрением крупных комплексов программ в народное хозяйство и оборонные системы. Рост доверия к программным продуктам и к их возможностям выполнять различные логические и вычислительные функции не только увеличил размеры разработок, но и значительно ***повысил важность и ответственность выполняемых ими функций.*** Возрастание требований к результатам функционирования сложных комплексов программ управления и обработки информации вызвало интерес заказчиков, пользователей и разработчиков к анализу факторов, определяющих ***стоимость и качество*** создаваемых и эксплуатируемых программ. Поэтому одной из основных задач исследований по проекту ПРОМЕТЕЙ в 1970-е – 80-е годы стали ***технико-экономический анализ и обоснование необходимых ресурсов для создания программных продуктов*** в соответствии с требованиями контрактов [12, 20].

При начале проектирования крупных программных продуктов, требующих заведомо ***больших экономических, трудовых и временных затрат,*** необходима была хотя бы приближенная, формализованная их оценка по определенной методике. Для этого должен был подготовлен согласованный между заказчиком и разработчиком первичный документ, в котором определены ***цели и задачи проекта, предполагаемые характеристики продукта и необходимые ресурсы для его реализации.*** Эти данные должны были предварительно

сбалансированы, и обеспечивать реализацию целей проекта при выделенных ресурсах с минимальным допустимым риском. Однако масштабы целей и функций сложных проектов имеют устойчивую тенденцию изменяться и ***увеличиваться по мере развития,*** а первоначально выделяемые ресурсы не удовлетворять их реализацию. Технико-экономическое обоснование проектов на начальном этапе их развития должно было содержать ***оценки рисков*** реализации поставленных целей, обеспечивать возможность планирования и выполнения жизненного цикла программного продукта или указывать на недопустимо высокий риск его реализации и целесообразность прекращения разработки.

Большую часть рисков и негативных последствий проектирования в 1970-е – 80-е годы можно было бы избежать, используя существовавшие, достаточно точные ***методы оценивания и прогнозирования затрат, а также управления проектами программных продуктов,*** для их успешного завершения. Эти последствия объяснялись многими причинами, из которых наиболее важными, являлись следующие:

• исходные тексты программных компонентов различны и по отдельности не полностью определяют сложность и размер конечного продукта;

• разработка сложных комплексов программ требует творчества и сотрудничества разных специалистов, индивидуальное и групповое поведение которых, как правило, трудно предсказать;

• в области экономики жизненного цикла программ накоплен относительно небольшой опыт количественных оценок, и его трудно было увеличивать, не обобщая разрозненные экономические эксперименты.

В 80-е годы в [15] было установлено, что в жизненном цикле сложных комплексов программ для обеспечения их высокого качества, ***целесообразно выделять специалистов, ответственных за анализ и прогнозирование экономических характеристик проектов.*** Необходимо было научить специалистов анализу и оцениванию конкретных факторов, влияющих на характеристики функционирования программных продуктов со стороны реально существующих и потенциально возможных негативных воздействий, и ограничений ресурсов проектов. Это должно привести в дальнейшем к созданию новой области экономической науки – ***экономики жизненного цикла программных продуктов.***

5.5. Сбор и обобщение экономических характеристик о жизненном цикле программных комплексов реального времени в 1980-е годы

Имевшиеся модели были не всегда столь точны, как хотелось бы, но могли весьма существенно помочь при технико-экономическом анализе и обосновании принципиальных решений создания сложных комплексов программ. ***Необходимы были дальнейшие, активные исследования на разных уровнях детализации,*** начиная от экономики и планирования создания программных продуктов в масштабах страны или предприятия и кончая экономикой выполнения частных операций отдельными специалистами при разработке или производстве конкретных компонентов и комплексов программ. Для этого программные продукты, и все процессы взаимодействия должны были связаны системой экономических и технических характеристик, в той или иной степени, использующих основные экономические показатели – ***реальные затраты ресурсов: финансов, труда и времени специалистов на конечный продукт.***

Только на базе серьезных статистических исследований технико-экономических показателей жизненного цикла и практического маркетинга ***возможны были обобщения и создание теоретических и практических основ экономики программных продуктов.*** Специалистам необходимо было исследовать и сформулировать методы технико-экономического анализа, оценивания и прогнозирования необходимых ресурсов для проектов разработки комплексов программ. Тем самым определить очень важный, базовый раздел из всей экономики программных продуктов. Такое выделение определялось тем, что без подобных базовых исследований вряд ли возможно было последующее серьезное развитие экономики в этой области. Внимание было сосредоточено на концептуальной основе распределения затрат труда в процессе разработки программ, на ***факторах, определяющих реальные трудозатраты и другие технико-экономические показатели,*** а также на исследовании таких ***характеристик*** в реализованных разработках комплексов программ.

Рассматривались преимущественно средние и крупные проекты, создаваемые коллективами специалистов. В таких проектах на чистое творчество, искусство и научные

исследования отдельных специалистов, преобладающие в небольших индивидуальных разработках, накладывается множество работ, характерных для индустриального проектирования и производства программных продуктов. Вследствие этого значительно ***нивелировались индивидуальные особенности отдельных специалистов,*** и появлялась возможность оценивать усредненные характеристики производительности труда и другие технико-экономические показатели (ТЭП) в крупных коллективах. Анализируемые процессы были ограничены: от этапа оформления требований технического задания на разработку продукта и до этапа завершения испытаний опытного образца программного продукта соответствующего требованиям заказчика.

Индустриальное, коллективное создание крупных программных комплексов необходимо было ***структурировать, планировать и поэтапно регламентировать*** с учетом заданных ограничений сроков и стоимости проектов. Это было обусловлено, в частности, предварительным отсутствием у специалистов количественных оценок ТЭП разработки программ на основе статистических данных по реально созданным программным продуктам. Задача состояла в том, чтобы детально исследовать реальные технико-экономические показатели достаточно представительного набора завершенных проектов комплексов программ и создать на этой статистической базе ***методики прогнозирования трудоемкости, длительности и числа необходимых специалистов*** по этапам работ и интегрально по проектам разных классов и размеров.

С этой целью в конце 70-х годов были разработаны методические указания и анкета, разосланные ***по приказу министерства радиопромышленности СССР*** в ряд оборонных предприятий для сбора сведений о каждой завершенной промышленной разработке программного продукта (Алексей Иванович Потапов) [20]. ***В анкетах подлежало регистрировать***:

- характеристики объекта производства – программного продукта;
- трудовые и временные ресурсы, непосредственно затраченные на разработку программного продукта;
- программная оснащенность разработки средствами автоматизации технологии производства;
- аппаратурная оснащенность разработки программ средствами вычислительной техники.

При этом фиксировались ***основные составляющие затрат на разработку комплекса программ.***

трудоемкость и длительность непосредственного

производства программного продукта;

• затраты на программные системы автоматизации разработки программного обеспечения (САР-ПО);

• затраты на аппаратуру ЭВМ, используемых в процессе разработки комплекса программ.

При расчете производительности труда разработчиков учитывался ***труд непосредственного создания комплекса программ,*** а при расчете стоимости разработки программного продукта в целом и одной команды – все составляющие затрат. Относительно небольшие затраты на системы автоматизации разработки программ не учитывались, так как были обусловлены многократным применением одних и тех же САРПО для создания различных проектов на определенном предприятии. Так же по многим проектам распределялись капитальные затраты на покупку и установку технологических ЭВМ. Поэтому при анкетировании не выделялись и не учитывались эти небольшие затраты, так как в этой составляющей доминирующими оказались затраты на машинное время ЭВМ.

Технико-экономические показатели проектов комплексов программ начали анализироваться в конце 1970 – х годов. Наиболее систематично они исследовались за период от 1982-го года до 1986-го года, когда были собраны, обработаны и проанализированы данные, поступившие с 30 оборонных предприятий министерства, о разработке 142 проектов программных продуктов реального времени, объемом около 10,5 млн. команд. Основными ТЭП проектов являлись как абсолютные показатели (размер, трудоемкость и длительность разработки), так и относительные (производительность труда разработчиков, затрата машинного времени на одну команду комплекса, стоимость разработки одной команды программного продукта). Относительные ТЭП дали возможность сопоставить различные разработки продуктов и коллективы разработчиков по эффективности их труда.

За меру производительности труда разработчиков принималось отношение объема созданных компонентов и комплекса в командах ***нового исходного текста программ*** (без учета заимствования из предшествовавших проектов) ***к трудозатратам*** на его разработку в человеко-днях. Учитывался только объем программного продукта, прошедшего испытания и переданного на эксплуатацию потребителю без экспериментальных компонентов и версий, не вошедших в окончательный программный продукт. При расчете трудозатрат принималась полная длительность производства (от разработки общего технического задания до сдачи продукта заказчику). В

число специалистов-разработчиков включались различные категории специалистов: руководители разработки, аналитики, системотехники, программисты и тестировщики, а также технический персонал, занятый оформлением текущей и итоговой документации.

В результате обработки собранных фактических статистических данных были получены ***зависимости ТЭП от основных факторов,*** выявлена роль применения различных языков программирования, методов доступа к ЭВМ, определена реальная обеспеченность разработчиков дисплеями и машинным временем технологических ЭВМ. По программам реального времени к 1986-му году ***производительность труда*** специалистов составила до 2,6 команд (строк текста) на человека в день по сравнению с 1,6 в 1980-м году. Особенно производительность труда возрастала в последние четыре года, что составило прирост, равный почти 50 %. Из общего объема разработок около 23 % составляли средние комплексы программ (объемом от 30 тыс. до 100 тыс. команд). К 1980-му году крупные комплексы (объемом более 100 тыс. команд) составляли 32 %.

Данные по производительности труда при разработке продуктов реального времени в начале 80-х годов, были сопоставлены с аналогичными данными в США. При выборке проектов с приблизительно одинаковыми характеристиками, производительность труда отечественных специалистов оказалась ***близка*** к доступным аналогичным данным специалистов за рубежом.

Важнейшим технико-экономическим показателем разработки программных продуктов является ее ***продолжительность.*** В 1980-м году средняя длительность разработки крупного комплекса программ реального времени была – 4 года; в 1985-м году длительность сократилась в среднем на 30 %. Факторами, снижающими длительность разработки, явились:

• повышение производительности труда разработчиков;

• применение готовых переносимых программных компонентов;

• уменьшение объемов разрабатываемых комплексов программ.

Выявлено, что наиболее эффективным направлением для улучшения ТЭП разработки программных продуктов являлось ***применение готовых программных компонентов.*** Анализ проектов показал, что при заимствовании от 30 до 80 % конечного объема продукта, производительность труда возрастала в диапазоне от 1,6–3,0 раза по сравнению со средней на предприятии. Заимствование 50 % готовых программ (по

общему размеру) позволило сократить длительность разработки, в среднем, на 25 %, а при заимствовании в 80 % – в два раза.

5.6. Технико-экономический анализ и обоснование разработки комплексов программ – 1985-е годы

Основная задача состояла в выборе и прогнозировании наиболее адекватных экономических и функциональных критериев для обобщенного описания эффективности, стоимости создания и использования комплексов программ в зависимости от их назначения, области применения и других факторов. Применение программных средств как ***продукции*** существенно повысило ***актуальность технико-экономического обоснования и прогнозирования*** их характеристик и процессов производства. Основной целью создания многих программных продуктов являлось повышение эффективности научных исследований, производства промышленных продуктов или управления объектами и системами реального времени, в которых применялись крупные комплексы программ. Такими системами могли быть средства автоматизированного управления прокатными станами, самолетами, ракетами, электростанциями, информационно-справочными системами административного управления и т. п. В ряде случаев программные продукты было невозможно или очень трудно характеризовать непосредственной экономической эффективностью. Примером могут служить программные продукты в системах управления воздушным движением или космическими аппаратами, а также в системах оборонного назначения или автоматизации научных экспериментов. В таких случаях при экономическом анализе программ невозможно было определять изменение прямой эффективности систем в зависимости от затрат, и целесообразно было из анализа исключать характеристики полной экономической эффективности и сопутствующие ей функциональные критерии качества [25]. Тогда исследование эффективности программного продукта можно было проводить, ***минимизируя затраты на разработку,*** в предположении, что полностью обеспечены заданные функциональные характеристики. При таком анализе должны были учитываться следующие цели [29].

Первая цель состояла в ***определении реальных затрат*** на разработку определенных компонентов и программных продуктов в целом, с учетом их сложности и требуемого качества. Для этого должна была изучена существующая практика разработки программ и обобщены ТЭП успешно завершенных проектов. Такие

обобщения должны были выявить трудоемкость (стоимость) и производительность труда при разработке реальных программ определенных классов и назначения, а также основные факторы, влияющие на эти показатели при создании конкретных комплексов программ. Кроме того, необходимо было определить длительность всего процесса разработки программ и его отдельных этапов. Для этого должны быть разработаны и внедрены методики сбора первичных техникоэкономических данных и их обработки, по завершенным или находящимся в процессе разработки проектам. В результате могли быть получены опорные значения основных ТЭП создания комплексов программ разных классов.

Вторая цель – создание методов и методик прогнозирования затрат и длительности разработки комплексов программ. Методики должны были учитывать полученные значения ТЭП, основные характеристики создаваемых комплексов программ, а также технологию, оснащенность и организацию их разработки. Получаемые прогнозы должны позволять эффективно планировать разработки, управлять созданием программ и осуществлять проекты в соответствии с заданными требованиями, сроками и затратами на основе анализа аналогов – прототипов.

Третьей целью анализа являлось ***обоснование и создание методов и средств снижения совокупных затрат и сроков*** разработки сложных программных продуктов. При этом возникали задачи:

- эффективного распределения общих трудовых ресурсов, используемых при производстве программ;
- развития и повышения экономической эффективности технологий, применяемых для создания программ различных классов;
- рационального повышения комплексной автоматизации технологий разработки программных продуктов;
- выбора методов и инструментальных средств, в наибольшей степени способствующих снижению длительности создания и совокупных затрат на программные продукты, а также повышению их качества.

Четвертой целью технико-экономического исследования процессов разработки программ являлось ***создание методических и нормативных документов,*** как основы планирования промышленной разработки аналогичных программных продуктов. ***Наличие нормативов*** могло коренным образом изменить характер разработки и приблизить его к отрасли ***современного регламентированного промышленного проектирования и***

производства. В результате появлялась возможность управления затратами на разработку, количеством и качеством создаваемых комплексов и их компонентов на различных этапах.

На практике классы систем при анализе обычно имеют ряд близких по значимости целей применения, и соответствующих им характеристик качества. В результате эффективность технологических решений приходится оценивать одновременно по нескольким показателям. Для этого следует стремиться сформулировать обобщенную скалярную функцию эффекта и затрат или строить нормированный вектор показателей качества. Во многих случаях эффективность сложной новой техники и комплексов программ в процессе проектирования приходилось прогнозировать ***в условиях неопределенности*** целей, различных факторов и характеристик. Обычно бывали недостаточно известны перспективы внедрения и эксплуатации объектов разработки – новых программных продуктов. Трудно формализуемыми и оцениваемыми являются размеры (масштабы) и структура систем, взаимодействие основных подсистем, цели, функции и критерии оценки эффективности их функционирования. Значительные неопределенности содержались также в технико-экономических характеристиках технологий, а также инструментальных средств автоматизации проектирования и изготовления программ.

Достаточно обычными в 1970-е – 80-е годы стали комплексы программ размером 100 и более тыс. строк текста. Возникла ***проблема декомпозиции комплексов программ*** на компоненты меньшего размера и сложности, а также упорядоченного структурно-иерархического построения таких программ. Известные методы снижения суммарной сложности систем путем декомпозиции, структурирования, локализации функций и упрощения компонентов нашли дальнейшее развитие при создании современных методов разработки сложных программных продуктов. Такие сложные программы в разумные сроки (1–5 лет) могли быть созданы только организованными коллективами специалистов под единым руководством. Размеры коллективов разработчиков и их структура в значительной степени диктуются структурой, функциональными задачами и характеристиками комплексов программ. ***Коллективность разработки*** привела к необходимости строгого планирования и формализации процесса разработки аналогично процессу проектирования других сложных промышленных изделий.

Программы реального времени обычно длительно функционируют во многих экземплярах системы. Это приводит к необходимости ***эффективного использования ограниченных***

вычислительных ресурсов, памяти и производительности ***реализующих (объектных) ЭВМ*** для функциональных программ, которые предназначены для решения основных задач по обработке информации или управлению. Включение комплексов программ в контур управления производством или динамическими объектами приводило к необходимости обеспечения ими высокого качества решения задач, ***безопасности и надежности их функционирования*** не ниже, чем у аппаратных компонентов соответствующих систем. Поэтому программы должны были тщательно отлаживаться и проходить контрольные испытания для определения уровня отлаженности, безопасности и надежности их функционирования. Отладка программ на технологических ЭВМ позволяла в ряде случаев значительно сокращать сроки разработки и проводить ее независимо от разработки, реализующей ЭВМ, для которой предназначен комплекс программ. ***Комплексное использование универсальных ЭВМ в качестве технологических и моделирующих*** позволяло успешно решать задачу создания надежных отлаженных программ для систем реального времени.

5.7. Методики технико-экономического обоснования производства комплексов программ – 1985-е годы

На базе выполненных исследований было разработано методическое руководство для оценки и согласования с заказчиками ТЭП и стоимости разработки крупных комплексов программ, что позволило сократить конфликты при подготовке договоров на создание программных продуктов в конце 80-х годов. Методика прогнозирования технико-экономических показателей разработки программных средств ***была, одобрена в*** 1990 году Ученым советом Центрального бюро нормативов по труду Госкомтруда СССР и рекомендована для применения в научно-производственных объединениях, на предприятиях и в организациях науки и научно-технического обслуживания отраслей народного хозяйства. На основе этой методики был разработан технологический пакет прикладных программ ПЛАПС, предназначенный для автоматизированного прогнозирования технико-экономических показателей разработки программных средств, а также обеспечивающий функции планирования и методической поддержки проектов.

Наиболее полные исследования, обобщения и экономические характеристики реализованных проектов за рубежом отражены в книге Б. Боэма [15, 11], которая стала доступной отечественным специалистам в середине 80-х годов. Приступая к разработке комплекса программ, как в любой промышленной деятельности, в методиках рекомендовалось проводить реалистическую оценку возможного ***масштаба проекта —*** поставленных целей, ресурсов проекта и выделенного времени. Задача управления масштабом состояла в задании базовых требований, которые включали разбитое на компоненты ограниченное множество функций и требований, намеченных для реализации в конкретной версии проекта. ***Базовый уровень масштаба, должен был обеспечивать*** приемлемый для заказчика минимум функций и требований к продукту, а также разумную вероятность успеха проекта с точки зрения возможностей коллектива разработчиков.

При оценивании масштаба следовало определить приоритеты реализуемых функций для установления состава работ, согласованного между заказчиком и разработчиком, которые ***обязательно*** должны быть выполнены и для определения базового уровня масштаба конкретного проекта с допустимым риском неуспешной реализации. В общем случае необходимо

было достигать сбалансированного состава целей оценивания разных характеристик, которые давали бы примерно одинаковую величину уровня неопределенности для всех компонентов комплекса. Кроме того, каждая оценка ТЭП должна была сопровождаться, указанием степени ее неопределенности. По мере разработки проекта их необходимо было пересматривать и изменять, когда это становится выгодным. Для этого в 80-е годы рекомендовались ***методики*** [29]:

• первичная оценка ТЭП при подготовке концепции и технического задания на новый комплекс программ, на основе экспертных данных размера комплекса, производительности труда или стоимости разработки одной строки текста программ – прототипов;

• прогнозирование ТЭП при детальном проектировании комплекса программ на базе расчетных значений трудоемкости и длительности его разработки с учетом влияния различных факторов.

В ***первой методике*** был реализован метод прогноза ТЭП с учетом ***экспертной оценки минимального числа факторов.*** Данная методика могла применяться, когда определены цели и общие функции проекта, сформулированы концепция и первичные требования с достоверностью около 30–40 %. Основная ***цель оценки ТЭП —*** подготовить возможность принять обоснованное решение о допустимости дальнейшего продвижения проекта в область ***системного анализа, разработки требований и предварительного проектирования.*** Если оказывалось, что рассчитанные технико-экономические показатели и требуемые ресурсы не могут быть обеспечены для продолжения проекта, то возможны были кардинальные решения: либо изменение некоторых требований, ТЭП и выделяемых ресурсов, либо прекращение проектирования данного продукта. Учитывая полноту и достоверность доступных характеристик и требований к проекту продукта должны были определены ***цели и возможная достоверность технико-экономического обоснования затрат на продолжение проектирования и производства программного продукта.*** При первичном технико-экономическом обосновании сложных проектов программных продуктов наибольшее значение имели ***три ключевых фактора***:

• размер – масштаб, подлежащих разработке полностью новых программных компонентов;

• размер и относительная доля готовых программных компонентов, которые могут быть заимствованы из предшествовавших проектов и повторно использованы в новом программном продукте;

• относительные затраты ресурсов на создание проекта: труда специалистов, времени и бюджета на единицу размера (на строку текста программ) создаваемого продукта.

Эти факторы могли быть оценены квалифицированными экспертами на основе имеющегося у них опыта реализации предшествовавших подобных проектов, а также использования опубликованных, статистических данных. Достоверность прогнозов ТЭП зависела, прежде всего, от ***точности экспертной оценки исходных данных,*** размера – масштаба комплекса программ, от достоверности экспертной оценки производительности труда специалистов или от стоимости разработки одной строки текста программы.

Основными исходными данными для оценок ТЭП являлись ***концепция проекта и комплекс требований*** к иерархическому набору функций, которые могут быть разбиты на предполагаемые фактические компоненты. В дальнейшем разбиение могло детализироваться, формируя упрощенный или более точный уровень абстракции и взаимодействия компонентов. Наиболее глубокий уровень детализации, как правило, редко формируется ко времени первоначальной экспертной оценки размера комплекса.

Экспертная оценка удельных затрат на строку текста программ в методике приводилась для некоторых примеров программных продуктов. При этом подчеркивалось, что оценки относятся к полному циклу разработки крупных комплексов программ, начиная от создания концепции и требований до завершения испытаний, и передачи программного продукта заказчику или пользователям. В составе участников проекта учитывались все категории специалистов, обеспечивавших реализацию программного продукта.

Так как разработчики сложных комплексов программ были ***не заинтересованы раскрывать реальную экономику выполненных проектов,*** и эти данные склонны относить к коммерческой тайне, опубликованные ТЭП носят отрывочный, не упорядоченный и не всегда достоверный характер. Кроме того, обычно не представлялись детальные сведения об особенностях и требованиях к качеству объекта разработки, применявшейся технологии и инструментальных средствах, характеристиках коллектива специалистов и других факторах. Поэтому такие данные трудно было обобщать и использовать для прогнозирования новых проектов.

В исследованиях [15] при разработке программных продуктов реального времени в 80-е годы, преимущественно на ассемблере, была получена средняя производительность 60–80 строк на

человеко-месяц. Для относительно небольших комплексов программ административных систем (в значительной степени на языках высокого уровня) производительность составила около 260 строк на человеко-месяц. Таким образом, в зависимости от характеристик объекта и условий разработки был возможен экспертный выбор величин производительности труда для последующего прогноза полной трудоемкости создания программных продуктов.

Экспертная оценка, длительности разработки сложного программного продукта могла оцениваться на базе рассчитанной ранее трудоемкости разработки, от которой нелинейно зависит длительность. Например, крупные продукты реального времени размером около 500 тысяч строк требовали для реализации около 3,5 лет, а небольшие (30 тысяч строк) – около одного года.

Экспертная оценка необходимого числа специалистов рассчитывалась путем деления полной трудоемкости разработки на длительность ее реализации. Для ***примера*** крупного продукта реального времени, размером 500 тысяч строк, необходимое число специалистов достигало 160 человек [15], а для относительно небольшого проекта (30 тысяч строк) – в десять раз меньше (16 человек). Аналогично можно было получить оценки необходимого числа специалистов на выделенных крупных этапах разработки, что полезно для первичного формирования коллектива и оценки возможности реализации ими конкретного проекта.

Сбор и обобщение экономических характеристик отечественных [20] и зарубежных [15] крупных комплексов программ в конце 80-х годов позволил существенно повысить достоверность прогнозирования технико-экономических показателей проектов. Это стимулировало разработку более точных моделей с учетом многих факторов, влияющих на оценки трудоемкости, длительности и числа специалистов, необходимых для разработки программных продуктов. В методике учитывались следующие ***группы дополнительных факторов и их детализации*** [20]:

- характеристики проектируемого программного продукта;
- квалификация коллектива разработчиков;
- технологическая среда разработки;
- организация проектирования и производства.

В качестве ***характеристик комплекса программ*** применялись:

- доля используемых готовых программных компонентов;
- требования к надежности (наработке на отказ);
- ограничения на доступные вычислительные ресурсы –

уровень (процент) загрузки ЭВМ;

• ожидаемая длительность сопровождения и эксплуатации версий программного продукта;

• ожидаемый тираж программного продукта;

• размер базы данных, с которой взаимодействует комплекс программ.

Группа ***характеристик коллектива разработчиков*** включала:

• тематическую квалификацию разработчиков (опыт работы в конкретной прикладной области, оцениваемый в годах);

• программистскую квалификацию разработчиков (опыт работы на конкретном языке программирования и с определенным инструментарием, оцениваемый в годах).

Группу ***характеристик технологической среды*** разработки составляли:

• быстродействие технологической ЭВМ, приходящееся на одного разработчика;

• размер – масштаб технологических программных средств;

• уровень используемого языка программирования (коэффициент расширения объектного кода программ в зависимости от уровня языка программирования).

К ***характеристикам организации процесса*** проектирования и производства относились:

• активность применения современных методов программирования (структурное программирование, структурное проектирование, бригада главного программиста и т. п.);

• стабильность исходных требований, представленных заказчиком в техническом задании.

Данные прогнозов по мере развития проекта могли уточняться и корректироваться с использованием реальных затрат ресурсов при последовательном завершении определенных этапов проекта. При этом возможен анализ технико-экономических показателей при изменении характеристик и разных вариантах состава функций и размера продукта, а также некоторых специфических факторов проекта, влияющих на ТЭП. При реальной разработке, тестировании и корректировке программ общая тенденция состоит в непрерывном ***увеличении размера*** комплекса, что при оценке ТЭП требовало отслеживания роста требуемых и используемых ресурсов. Кроме того, технико-экономические показатели, требующиеся ресурсы и планы разработки должны были корректироваться путем ***исключения*** из прогнозируемых ТЭП затрат ресурсов, уже использованных на завершенных этапах проекта.

Принципиальным отличием методики ПЛАПС являлась

возможность планирования процесса разработки комплекса программ, а также учета и корректировки реальных значений трудоемкости и длительности. В методике была реализована визуализация планов разработки комплекса программ в виде ***диаграмм Ганта*** по шести укрупненным этапам и множеству (около 40) частных работ, составлявших каждый из этих этапов. Состав и характеристики работ и этапов можно было корректировать по длительности и взаимодействию с предшествующими и последующими работами. При ***обобщении рассчитанных технико-экономических показателей*** на практике, некоторые значения, могли не удовлетворить специалистов, ведущих прогнозирование. Поэтому в методике допускалась ***возможность пересчета*** получаемых прогнозных значений на основе дополнительно вводимых значений некоторых факторов или ТЭП.

Глава 6. Примеры создания заказных систем на основе программной инженерии в 1960-е – 80-е годы

6.1. Технологии программной инженерии для крупных комплексов программ системы ПВО в 1960-е – 80-е годы

Примером одного наиболее крупных проектов вычислительных систем и ***комплексов программ реального времени*** оборонного назначения может служить ***начало разработки в 50-х годах*** системы противовоздушной обороны (ПВО) страны [11, 12]. В 1957-м году в НИИ-5 началось формирование подразделений, подбор и обучение сотрудников для того, чтобы задачи и системы обработки радиолокационной информации, а также наведения и управления истребителями-перехватчиками перевести на алгоритмы и программы для цифровой вычислительной техники территориальных командных пунктов. Эти первичные анализы и проработки послужили ***прототипами*** для создания впоследствии сложной глобальной телекоммуникационной сети радиолокационных узлов ***(РЛУ)*** (см.п. 4.3). В это же время на машине «Урал – 1» в институте началось моделирование алгоритмов обработки радиолокационной информации. Моделировались алгоритмы сопровождения движущихся воздушных объектов в ***имитированном псевдореальном времени.*** На машине с производительностью 100 операций в секунду и общей памятью две тысячи слов, занимавшей огромное помещение, впервые начинали ставиться и решаться сложные комплексные ***задачи РЛУ с имитацией реального времени.*** Моделировались исходные координаты, отражающие движение наблюдаемых радиолокаторами объектов, которые могли составлять группы и маневрировать. По этим данным имитировалось обнаружение динамических объектов; формировались траектории их движения, отождествление и объединение траекторий динамических объектов в реальном времени от различных РЛУ. В 1959 году в институте появилась более мощная машина М-20, моделирование было продолжено на ней и расширено. У руководства института появлялась уверенность, что подобные задачи можно решать на цифровых ЭВМ по данным радиолокационных станций, взаимодействующих в сети на большой территории.

В 1959 году в НИИ-5 был разработан проект территориальной, распределенной информационной системы ПВО (генеральный конструктор Анатолий Леонидович Лившиц) [12], которая в современной терминологии ***имела, следующие особенности*** (опытный образец в полном составе испытан в 1969

году):

• территориально-распределенная информационная система на ЭВМ с многими пунктами сбора и обработки радиолокационной информации и командными пунктами управления активными средствами ПВО – ракетами и истребителями-перехватчиками;

• радиолокационная информация о воздушных целях от различных источников из зоны их обнаружения, обобщалась на командных пунктах, что обеспечивало непрерывность траекторий целей для возможности действия активных средств ПВО;

• все средства обработки информации и управления должны были работать на ЭВМ в реальном масштабе времени при несинхронных потоках сообщений от источников информации с временем отклика измеряемом долями секунды;

• на каждом пункте обработки радиолокационной информации и управления должны были применяться, объединенные в локальную сеть графические терминалы различных типов (характроны, телевизионные индикаторы) для визуализации воздушной обстановки и обеспечения функционирования оперативного и командного состава;

• вычислительные средства обработки радиолокационной информации и командных пунктов имели очень ограниченные ресурсы по памяти и производительности и низкую надежность и, тем не менее, должны были эффективно и надежно решать заданные функциональные задачи ПВО;

• телекоммуникационные каналы связи работали по унифицированным протоколам, имели относительно низкую пропускную способность (телефонные каналы) и большой процент ошибок в сообщениях.

Командные пункты управления активными средствами ПВО были построены на стационарных, полупроводниковых ЭВМ «Радон» (главный конструктор Сергей Алексеевич Крутовских) и комплексах расширения памяти «Кристалл» (впоследствии на ЭВМ «Гранит»). Сотни тысяч команд в программах командного пункта были разработаны в машинных кодах под руководством Залмана Михайловича Бененсона к 1968-му году, а затем развивались и совершенствовались многие годы. В разработке алгоритмов и программ участвовало свыше двухсот человек.

Система ПВО базировалась на множестве подвижных ***радиолокационных узлов (РАУ),*** образующих почти сплошное поле радиолокационного обнаружения в стране. Для автоматизированной системы обработки информации радиолокационного узла ***«Межа»*** использовалась ***мобильная ЭВМ 5Э89*** (см. п. 2.6). На машине выполнялась обработка

информации, поступающей в реальном масштабе времени от радиолокаторов кругового обзора, и автоматизированное сопровождение воздушных целей, истребителей и ракет. Разработка опытного образца системы ПВО была успешно завершена в 1968-м году, и началось ее серийное производство. Разработка программ РЛУ была очень трудоемкой и показала необходимость автоматизации проектирования и производства комплексов программ реального времени.

Концепция технологии программной инженерии была развита и апробирована в середине 70-х годов при проектировании и внедрении в ПВО версий системы автоматизации разработки программного обеспечения (САРПО см. п 3.5) и комплексных испытательных моделирующих стендов (КИМС), на предприятиях различных отраслей оборонной промышленности (см. главу 4). Ее основные положения первоначально использовались и испытывались при создании инструментальных систем САРПО, а также на их основе в течение всего жизненного цикла ряда систем оборонного назначения. С использованием этих САРПО были разработаны и сопровождались программные продукты с общим объемом в ***несколько десятков миллионов команд.***

Конструктивные характеристики, и особенности функционирования объектов разработки оборонных систем, принятые при создании ПРОМЕТЕИ – технологии, отличались от других комплексов программ в следующем:

• высокая информационно-логическая и структурная сложность комплексов программ;

• большой размер (объем) создаваемых комплексов программ – до нескольких сотен тысяч операторов – строк текста программ;

• большая размерность – число входящих компонентов (модулей) до нескольких тысяч;

• разнообразие обрабатываемых типов и единиц данных, число которых могло доходить до нескольких десятков тысяч;

• наличие глубокой связности программных компонентов по информации и управлению и необходимость их комплексирования в целостный комплекс программ;

• высокая логическая сложность программных компонентов – среднее число команд принятия решений (альтернатив) в программах – одно на 5 – 10 команд;

• основной способ запуска функциональных программ для исполнения – автоматический от информации, поступающей от объектов внешней среды или от других программ в случайные моменты времени;

• полное и глубокое документирование компонентов и

комплексов программ, продолжительное конфигурационное управление и сопровождение версий программных продуктов.

Особенности функционирования создаваемых комплексов программ, определившие ПРОМЕТЕЙ-технологию в целом и характеристики инструментальных систем автоматизации, включали:

• ограниченность ресурсов памяти и производительности специализированных ЭВМ, ориентированных на решение ***целевой, функциональной проблемы*** конкретной системы обработки информации и управления;

• требования малого времен реакции – отклика (миллисекунды или секунды) на поступающую информацию от объектов внешней среды, и подлежащую обработке;

• большое число асинхронных, взаимодействующих с комплексом программ внешних источников и потребителей информации (десятки, сотни);

• высокие требования к качеству, надежности и безопасности функционирования комплекса программ, вследствие чего необходима высококачественная отладка и испытания в динамике в условиях имитированной и реальной внешней среды;

• набор штатных внешних устройств и диалоговых средств таких ЭВМ был ограничен и не позволял на них вести автоматизированную разработку комплексов программ;

• длительное зачастую непрерывное функционирование комплексов программ при эксплуатации (круглосуточно, годами).

Область применения технологии определила методы и средства автоматизации регламентированных процессов реализации комплексов программ, автоматизацию всех функционально связанных этапов и операций технологического процесса. В частности, это достигалось созданием общей базы данных проектирования, в которой хранились компоненты комплекса программ во всех формах представления (исходные спецификации, тексты программ на языке и в объектном коде, тесты, документы и т. д.). Сквозная, ***технология программной инженерии*** включала скоординированную автоматизацию всех этапов ЖЦ, как отработку комплекса программ на специально выделяемой технологической ЭВМ (кросс-технология), так и на специализированной мобильной ЭВМ, реализующей комплекс программ (резидент-технология). Для этого использовались ***следующие основные технологические принципы***:

• реализация технологических инструментальных систем для обеспечения автоматизации жизненного цикла, программирования и отладки комплексов программ, специализированных ЭВМ на универсальных машинах с

большими вычислительными ресурсами и единой базой данных проектирования;

• автоматизированная настройка трансляторов, интерпретаторов и всего унифицированного технологического инструментария на архитектуру различных специализированных ЭВМ и характеристики создаваемых программных продуктов реального времени;

• применение системы взаимосвязанных языков программирования, преимущественно уровня автокода и макроязыков, для обеспечения высокой эффективности программ по использованию памяти и производительности, специализированных ЭВМ в реальном времени.

Эффективность автоматизации процессов разработки и сопровождения в значительной степени определялась унификацией и упорядоченностью, как технологического процесса разработки, так и полнотой документирования проектируемого комплекса программ. Были регламентированы технологический процесс, состав, формы представления результатов и критерии качества выполнения последовательных этапов работ. Для этого была разработана совокупность взаимосвязанных методов, обеспечивающих регламентацию и автоматизацию жизненного цикла комплекса программ на всех этапах технологического процесса. Развитая методология учитывала, что ЖЦ комплекса программ связан с выполнением как творческих (разработка алгоритмов, распределение функций комплекса программ по компонентам, выбор структуры комплекса программ, тестов, отладка и т. п.), так и значительного числа нетворческих, рутинных операций (ввод текстов, сбор сводных данных, контроль выполнения формальных правил программирования и т. п.). Качество комплексов программ должно было обеспечено автоматизацией и контролем операций.

При создании средств автоматизации особое внимание было уделено организации ***комфортного и рационального диалогового общения*** пользователей-разработчиков со средствами автоматизации и наглядному представлению обобщающей информации специалистам. Разделение видов и объектов труда разработчиков и регламентация результатов этапов работ являлась основой для ***разграничения ответственности специалистов*** разной квалификации за качество конечного продукта – модулей, компонентов и комплексов программ. Этот принцип определял необходимость формализации требований к форме представления и качеству результатов каждого этапа работ в технологическом процессе. Долгосрочное и оперативное поэтапное планирование работ

коллектива на основе доступных ресурсов (трудоемкости, ресурсов технологической и специализированной ЭВМ) должно было обеспечивать автоматизированный контроль хода проектирования и оперативное корректирование планов с учетом этапа разработки, наличия и квалификации специалистов и имеющихся ресурсов.

Исследования по программе ПРОМЕТЕЙ позволили уточнить ряд положений концепции по организации коллективов специалистов при создании крупных комплексов программ реального времени (см. главу 4). В 60-е годы были предприняты организационные меры, ***алгоритмисты и программисты объединены в небольшие группы – «команды»*** для полного решения определенных ***функциональных задач***, входящих в крупные системы. Руководитель такого относительно небольшого коллектива – «команды» (5 – 10 человек), полностью отвечал за результаты и качество решения конкретной функциональной задачи системы и был обязан достаточно хорошо разбираться в программировании и в ее алгоритмах. Отдельный коллектив, наиболее квалифицированных специалистов – ***интеграторов*** выделялся для комплексирования набора программных компонентов ряда функциональных задач в целостный комплекс программ.

Модульно-иерархическое структурное построение комплексов программ должно было обеспечивать упорядоченное их построение, выделение компонентов и модулей, организацию и унификацию связей между ними. Одновременно этот принцип определял необходимость упорядочения внутренних структур программных модулей, массивов данных и комплексов программ в целом. Реализация таких структур комплексов программ и правил написания отдельных компонентов, должны были обеспечивать использование разработанных компонентов ***как комплектующих изделий*** в нескольких вариантах комплексов программ близкого функционального назначения, в том числе для различных специализированных ЭВМ. Раздельная компиляция модулей на основе упреждающей разработки базы данных информации комплекса программ обеспечивала первоочередную разработку описаний глобальных данных, хранение этих описаний в базе данных проектирования и организацию на этой основе независимой компиляции программных модулей.

Для динамической комплексной отладки и испытаний программного комплекса и систем реального времени был разработан (1980-е годы) и применялся моделирующий стенд автоматизированной генерации тестов при имитации

динамических объектов внешней среды – ИСТРА (Павел Гаврилович Гаганов, Андрей Николаевич Зубковский) [12, 11]. Стенд мог использоваться для испытаний систем и комплексов программ управления системами ПВО, а также: воздушным движением самолетов по трассам и в зоне аэропортов; испытанием бортовых систем гражданских и военных самолетов.

Комплексные испытательные моделирующие стенды (КИМС) были проблемно – ориентированы и объем программ, моделирующих в них внешнюю среду, зачастую ***превышал размеры*** соответствующих испытываемых программ (например, системы ПВО). Для их реализации выделялись достаточно мощные, по тем временам, универсальные ***моделирующие ЭВМ–*** БЭСМ-6 – АС 6. Кроме того, для автоматизации разработки программ использовались отдельные специализированные, ***технологические ЭВМ,*** что в совокупности образовывало ***комплексную инструментальную базу*** для обеспечения всего ЖЦ и имитации функционирования крупных комплексов программ реального времени на специализированных мобильных ЭВМ.

В частности, использовался набор моделирующих программ, имитировавших данные о полетах самолетов противника и траекториях движения истребителей-перехватчиков и ракет. При этом учитывались зоны обнаружения территориально-распределенных радиолокационных узлов и функционирование командных пунктов при перехвате и поражении целей. Моделирование осуществлялось с помощью средств КИМС, не входящих в состав тестируемых программных средств компонентов систем ПВО на специализированных ЭВМ. Это позволило вести разработку моделей внешней среды одновременно или до создания комплекса программ, подлежащего динамической отладке и испытаниям. Полнота и приемлемая точность имитации условий и компонентов внешней среды в КИМС обеспечивала проверку действий средств ПВО в условиях, максимально приближенных к требованиям заказчика на средства обнаружения целей и их перехвата.

Контроль хода эксперимента, управление режимами испытаний и оперативная корректировка параметров объектов внешней среды осуществлялась испытателями ***с автоматизированных рабочих мест,*** включавших в свой состав дисплеи, графопостроители, пульты оператора-испытателя. Ввод-вывод графической и алфавитно-цифровой информации мог осуществляться с помощью графических дисплеев испытателей.

В рассматриваемом примере КИМС, модели воздушной обстановки состояли, в основном, из двух групп: модели

движения самолетов противника и модели движения самолетов и ракет перехватчиков. На аппаратуру отображения КИМС выводилась информация о движении всех воздушных объектов и идентифицирующая их информация в реальном времени. В соответствии с исходными параметрами сценария эксперимента, моделировались траектории движение самолетов противника с учетом времени, зон и дефектов их обнаружения различными радиолокационными средствами. Эти данные в реальном времени из КИМС транспортировались в специализированную ЭВМ испытываемой системы ПВО, где обрабатывались для принятия решений.

На командном пункте по этим данным принималось решение о перехвате целей, и в КИМС передавалась исходная информация для моделирования движения истребителей или ракет перехватчиков. Их траектории генерировались и управлялись соответствующими моделями КИМС и при необходимости корректировались командами с пункта наведения через штатную аппаратуру обмена информацией с бортом самолета или ракеты перехватчика. Результаты моделирования перехвата обрабатывались и отображались на графических дисплеях специализированной ЭВМ и регистрировались на графопостроителях КИМС. Кроме того, обобщалась информация о характеристиках качества функционирования программных продуктов системы ПВО, о выявленных дефектах программ и ошибках операторов. Комплекс программ моделирования объектов внешней среды КИМС составлял около ***400 тысяч строк ня автокодах БЭМШ и АС-6.***

Использование КИМС для динамической отладки и испытаний крупномасштабных программных продуктов высокого качества по экспертным оценкам сокращал затраты и время на их отработку на 40–50 %. Это обеспечивалось за счет автоматизации технологии отладки и более качественной проверки и отработки программ на моделях. Сокращалось также в несколько раз требуемое число натурных экспериментов, потребность в экспериментах с привлечением реальных летных средств уменьшалась в среднем в 3–5 раз. Создание КИМС и его программных моделей потребовало значительных затрат материальных и интеллектуальных ресурсов. При оценке рентабельности разработки такого стенда необходимо было определять, как выигрыш от его применения, так и затраты на его разработку.

Системы ПРОМЕТЕЙ-технологии ***позволили повысить производительность труда*** при создании крупных комплексов программ реального времени в ряде предприятий оборонной

промышленности приблизительно на порядок, до 3–5 команд в день на человека. С использованием этих систем были разработаны, сопровождались и эксплуатировались программные продукты ***с общим объемом в несколько десятков миллионов команд.*** Все разработки комплексов программ реального времени базировались на оригинальных отечественных идеях, методах и инструментальных средствах. Во всех системах практически полностью отсутствовали зарубежные технологические компоненты (кроме ОС ЕС). В 1988-м году в силу ряда организационных обстоятельств и ***«смуты и разрухи»*** в стране ***работ по развитию программной инженерии,*** комплексной ПРОМЕТЕИ – технологии и перечисленных инструментальных систем ***прекратились.***

6.2. Создание и внедрение программных продуктов систем противокосмической обороны и морской космической разведки

Примером создания крупного комплекса программ реального времени в начале 60-х годов была разработка, испытание и внедрение оборонной системы для перехвата и уничтожения опасных космических аппаратов (КА) [13, 11]. При появлении космической тематики в КБ-1 создание системы противокосмической обороны (система «ИС») было поручено в 1960-м году отделу, руководимому Анатолием Ивановичем Савиным, который возглавлял многие последующие проекты в этой области в ЦНИИ Комета. Среди новых оказалась задача управления КА – перехватчиком для его наведения и поражения КА-агрессора на основе сложных алгоритмов обработки информации, с чем не могла справиться применяемая ранее аналоговая техника. Необходимо было ***включить в контур управления КА цифровую вычислительную машину.*** В это время на предприятии имелся наземный вычислительный центр из нескольких машин типа М-50 с техническим персоналом и программистами, среди которых были выпускники ведущих вузов с физико-математической подготовкой, которые выполняли текущую работу по формализации задач, разработке алгоритмов и отладке программ, а также по автоматизации программирования. Однако в данном случае необходимо было, чтобы машина могла производить необходимые вычисления в реальном времени, реагируя на внешние сигналы, поступающие по каналам связи, принимать и выдавать информацию в нужные моменты. Такого опыта не было. Требовалось создать специализированную вычислительную машину и реализовать на ней процесс, соответствующий предполагаемой временной диаграмме функционирования оборонной системы. Основные задачи состояли в следующем:

- измерение параметров движения цели – опасного космического аппарата агрессора по данным радиолокационных станций;
- прогнозирование движения космического аппарата – цели;
- расчет траектории наведения на цель космического аппарата перехватчика;
- определение параметров движения космического аппарата – перехватчика по данным радиолокатора;
- расчет параметров старта ракеты – перехватчика и

программы расчетов первого витка движения цели;

• подготовка и выдача информации на стартовую позицию КА для перехвата цели;

• наведение космического аппарата – перехватчика на цель методом трех импульсной коррекции его траектории.

Необходимость разработки программ реального времени с высоким качеством функционирования, надежности, ресурсными ограничениями по занимаемой памяти, временам реакций по разным задачам для еще несуществующей машины в заданные сроки привело к необходимости выполнения ряда научных работ (Владимир Федорович Гребенкин):

• разработке специального программного комплекса решения задач реального времени: диспетчеризации и планирования функциональных задач, контроля, обмена и резервирования;

• разработке кросс-систем отладки, позволяющих отлаживать программы на универсальной ЭВМ М-50 в архитектуре и формате команд управляющей, специализированной ЭВМ БШВЦ и поставлять на объект внедрения инсталляционный комплект перфокарт;

• разработке методов оценки трудозатрат на программные работы и прогнозированию сроков и количества исполнителей на основе статистики по имеющимся аналогам.

Все это явилось основой ***развития программной инженерии*** как метода создания специальной программной продукции. Программное изделие для системы «ИС» представляло собой комплект перфокарт (толщиной 30 см) и комплект программной документации. Общий объем разработанного программного обеспечения составлял – 40 тыс. слов, среди которого:

• функциональное программное обеспечение по орбитальным расчетам и обработке радиолокационных измерений – 32 тыс. слов

• системные задачи, явившиеся прообразом операционной системы реального времени: счет времени, организация запуска задач по запланированному времени, контроль вычислений с восстановлением при сбоях, диспетчеризация задач, сохранение и восстановление результатов вычислений, организация резервирования по информации и вычислительным машинам – 5 тыс. слов;

• средства отладки в виде кросс программ, позволяющих документировать работу отлаживаемой программы – 3 тыс. слов.

Для решения этих задач оказалось необходимым создание специальной аппаратно-программной системы, обеспечивающей мультипрограммную организацию вычислительного процесса. Повышенные требования по надежности и уровню автоматизации

были решены путем создания трех машинного комплекса с аппаратной организацией межмашинного обмена. Характеристики разработанной на предприятии специализированной ЭВМ БШВЦ были следующие:

• двухадресная система команд с использованием трех индексных регистров;

• 42-разрядная структура слова, из которых 2 контрольных;

• арифметика с плавающей запятой, 6 разрядов – порядок числа, 34 разряда – мантисса;

• объем ОЗУ – 32 тысячи слов;

• внешняя память на магнитном барабане объемом 50 тысяч слов;

• быстродействие – 50 тысяч простых операций в секунду;

• прямой канал обмена с получением информации от абонента в ОЗУ;

• мультиплексные каналы для обмена с внутренними абонентами.

Вы числительный центр из трех машин БШВЦ с дополнительным оборудованием занимал площадь 600 кв.м. и потреблял около 1000 кВт. В состав радиолокационной станции определения координат и передачи команд (СОК ПК) входила ЭВМ БШМ, обеспечивающая взаимодействие устройств СОК ПК. Сложности разработки ПО БШМ состояли в необходимости обслуживания устройств СОК ПК за интервал времени 0,3 сек. при имеющейся арифметике с фиксированной запятой.

С 1963-го года начали появляться первые алгоритмы по расчетам орбитальных параметров, команд коррекции, обработке измерений, прогнозу местоположения КА. Алгоритмы программировались и отрабатывались на универсальной ЭВМ М-50. Стали возможны оценки объемов, трудозатрат и сроков на программные работы. Выявилось, что срок готовности системы к натурным испытаниям определяется тремя годами с момента поставки управляющей ЭВМ. Это означало, что поставленные технические средства должны были ожидать окончания отладки программного обеспечения в течение трех лет. Этот факт способствовал быстрому внедрению кросс-системы отладки программ для БШВЦ на М-50. Исходные команды программ набивались на разработанном перфораторе в формате БШВЦ, ***отлаживались на интерпретаторе в системе команд БШВЦ на М-50*** с замедлением в 100 раз. Способ перевода отлаженных процедур в макрокоманды позволил отладить программы реального времени к моменту поставки макетного образца БШВЦ. Кросс-система функционировала в 1964-й – 1966-й годы и выполнила свое предназначение. С 1966-го года отладка

программ выполнялась непосредственно на объектах внедрения систем ИС и МКРЦ двумя лабораториями программистов по 15 специалистов. Размеры программных комплексов по современной классификации соответствовали «средним» проектам (30–40 тыс. строк). В дальнейшем процесс сопровождения комплексов программ были выполнены версии на персональных ЭВМ с одновременной модернизацией средств сопряжения с абонентами.

За короткий срок был создан наземный командный пункт управления, разработана аппаратура управления КА-перехватчика. После целого ряда успешных экспериментов 1 ноября 1968-го года впервые в мировой практике были осуществлены орбитальный перехват и кинетическое поражение КА-мишени. В общей сложности в ходе испытаний по поражению космических объектов было выполнено 7 натурных работ с положительными результатами, что подтвердило высокие тактико-технические характеристики системы «ИС». В 1973-м году она была принята на вооружение.

Практически одновременно с разработкой системы «ИС» выполнялась разработка системы морской космической разведки и целеуказания (УС), для которой использованы аналогичные технические средства и базовые структуры программного комплекса. Отличие от системы «ИС» состояло в составе задач функционального программного продукта. Система «УС» была принята на вооружение в 1975 году.

6.3. Крупные комплексы программ реального времени в системах предупреждения о ракетном нападении

В 1971-м году в «ЦНИИ «Комета» (Анатолий Иванович Савин, Виктор Порфирьевич Мисник) начались работы по созданию космической системы ***обнаружения стартов баллистических ракет*** «УСК» для предупреждения о возможности ракетного нападения (СПРН) [13, 11]. Группировка космических аппаратов с геостационарными орбитами наблюдала за постоянными районами. Кадр размером в пространстве 1000х1000 км. с возможными отметками факелов стартующих ракет с периодом четыре секунды сбрасывался на командный пункт, где выполнялась обработка информации с выделением сигнала и определением параметров возможной траектории стартующей ракеты. Задача решалась путем создания и комплексирования программно-технических средств, выполняющих самостоятельные функции, объединенные единой целью. Конструкторы наземных и бортовых средств применили различные вычислительные машины. Комплекс управления радиолинией применил трехмашинный вычислительный комплекс из прототипов ЭВМ ЕС (МСМ) с урезанным форматом данных и усложненной системой прерываний. Комплекс управления КА совместили с управлением радиолинией. В комплексе обработки информации применялась ЭВМ М-10 с векторной арифметикой, целесообразной для выполнения однотипных операций над массивами данных. Комплекс обработки телеметрической информации составлял двухмашинный вычислительный комплекс из ЭВМ МСМ. Бортовая специализированная ЭВМ, размещенная на КА (МБУ-03) и ЭВМ контроля и обслуживания КА на стартовой позиции (МК-100) разрабатывались на предприятии «ЦНИИ «Комета».

Общий объем программного комплекса для функционирования системы «УСК» составил около 300 тысяч строк исходных команд, большинство из которых разрабатывались в машинных кодах с простейшими средствами отладки. Кросс-системы применялись отдельными группами разработчиков для предварительной отладки компонентов программного комплекса, чаще всего библиотек стандартных программ. В разработке участвовали около 300 специалистов. Имевшийся на предприятии опыт, полученный до начала создания программных средств «УСК», позволил вести разработку методами, которые сейчас можно оценить, как начало

применения программной инженерии, осознаваемой и создаваемой в процессе производственной и научной деятельности во взаимодействии с другими организациями и прежде всего с МНИИПА при работах по НИР ПРОМЕТЕЙ (1980-е годы). Определились этапы разработки в виде: проектирование, программирование, комплексирование и испытания. Выполнялось планирование работ, основанное на предполагаемых трудозатратах, и управление на основе технических решений и распределении ресурсов, необходимых для разработки комплекса программ. Появилась покомпонентная структура программного обеспечения с выделением операционной системы. Определялось качество программ, в том числе надежность, определяемую ошибками, приводящими к нарушению функций и даже отказам.

Комплекс программ управления системой и радиолинией представлял собой множество объектов, которые в случайные моменты времени поставляли информацию в вычислительный центр (ВЦ). Информация должна была быть обработана к заданному моменту и выдана на объект. При нескольких десятках объектов организация процесса обслуживания представляла собой новую задачу, которая потребовала создания специализированной операционной системы реального времени (СОС РВ). Эта система была создана в виде комбинации трех систем массового обслуживания в составе: прием заявок на обслуживание; организация вычислений с динамическими приоритетами с учетом параметров каждого одиночного процесса; организация обмена с учетом пропускной способности каналов. Часть процессов определяла качество системы по обеспечению надежности процесса средствами контроля и резервирования.

Обработка информации отличалась высокой функциональной сложностью, связанной со стохастическими процессами, выделением сигнала среди помех, которые не поддавались математическому описанию, в связи с новизной предметной области (космос, факел наземной ракеты, аппаратура обнаружения, космический аппарат, динамика орбиты). Прием и обработка телеметрии характеризовалась разнообразными потоками сообщений, что потребовало решения задач по оптимизации времени и методов обработки. Проверка и контроль КА на стартовой позиции должны были выполнять полноценный контроль КА в период предстартовой подготовки, что обеспечивалось применением штатной и имитационной аппаратуры и проведением проверки работы в соответствии с реальной временной диаграммой функционирования КА на орбите. Объемы программных комплексов, обеспечивавших

функционирование системы (без учета технологических средств разработки программ и проверки технических средств) составляли:

• управления системой и радиолинией – 100 тыс. строк;

• обработки информации внешней обстановки – 100 тыс. строк;

• приема и обработки телеметрии КА – 50 тыс. строк;

• проверки и контроля КА на стартовой позиции – 30 тыс. строк;

• программы необслуживаемой бортовой системы КА – 4 тыс. строк.

В 1978-м году были завершены государственные испытания, и в 1979-м году она была принята на вооружение.

В 1980-м году начались работы по созданию глобальной космической системы обнаружения стартов баллистических ракет «УСК-МО». Группировка КА на стационарных орбитах должна была наблюдать за всеми ракетоопасными районами мира. Это потребовало создания новой системы, которая по основным компонентам входящих систем совпадала с «УСК», а по сложности решаемых задач превосходила ее в несколько раз. Комплекс управления КА базировался на ВК 3700, состоящий из трех ЭВМ 70Т6. Комплекс обработки информации составлял спецвычислитель разработки ЦНИИ «Комета» для предварительной обработки информации, ВК «Эльбрус» для определения параметров траекторий ракет и двух ЭВМ 70Т6 для формирования и выдачи информации внешним абонентам. Комплекс обработки телеметрической информации включал двухмашинный вычислительный комплекс из ЭВМ ЕС 1045. Бортовая ЭВМ, размещенная на КА была ЭВМ МБУ-3 разработки ЦНИИ «Комета». Комплекс контроля и обслуживания КА на стартовой позиции включал ЭВМ СМ 1425 и группу ЭВМ «Электроника».

Общий объем комплекса программ для обеспечения функционирования системы «УСК-МО» достигал 650 тыс. строк команд. Применялись средства автоматизации и языки высокого уровня. Кросссистемы разрабатывались и применялись отдельными группами разработчиков, прежде всего, для отладки встроенного комплекса программ для бортовой ЭВМ. В разработке программ участвовало около 500 специалистов. Успешному созданию программных комплексов способствовало последовательное формирование программной инженерии в ЦНИИ «Комета», во многом благодаря участию в Координационном совете Министерства радиопромышленности СССР по автоматизации проектирования программного

обеспечения и в НИР ПРОМЕТЕЙ по разработке автоматизированных технологий создания крупномасштабных программных средств для систем реального времени.

Работы ЦНИИ КОМЕТА проводились по созданию сквозной автоматизированной технологии разработки алгоритмического и программного обеспечения для необслуживаемых специализированных бортовых вычислительных комплексов космических аппаратов (КА) системы предупреждения о ракетном нападении [9, 11]. Эта технология базировалась на инструментальных ЕС ЭВМ, СМ ЭВМ и комплексных имитационно-моделирующих стендах, которые кроме инструментальной среды содержали реальные бортовые вычислительные комплексы, а также некоторые элементы аппаратуры управления КА и функционировали в реальном времени. Технологические средства ПРОМЕТЕЙ-технологии, созданные в 80-е годы в ЦНИИ КОМЕТА, непрерывно совершенствовались, переносились на другую современную инструментальную среду.

В «ЦНИИ «Комета» были выполнены научно-исследовательские и опытно-конструкторские работы по направлениям программной инженерии:

• исследование жизненного цикла разработки сложных программных комплексов, анализ статистики ошибок, создание автоматизированных технологий отладки, обеспечивающих качества программных комплексов оборонного назначения;

• исследование и разработка средств для динамической отладки и испытаний программ объектных ЭВМ в реальном масштабе времени с использованием комплексных испытательных моделирующих стендов;

• исследование методов организации вычислительных процессов в реальном времени, создание специализированных операционных систем управления и обработки информации;

• исследование экономики разработки крупных программных средств, методов планирования и оценки ресурсов для их создания;

• обеспечение качества, стандартизации и сертификации сложных программных комплексов.

Объемы программных комплексов, обеспечивших функционирование системы (без учета технологических средств разработки программ и проверки технических средств), составили:

• комплекс управления радиолинией – 50 тыс. строк;
• комплекс управления системой – 200 тыс. строк;

• комплекс обработки информации – 200 тыс. строк;
• комплекс приема и обработки телеметрии – 100 тыс. строк;
• комплекс проверки и контроля КА на стартовой позиции – 50 тыс. строк;
• комплекс программ бортовой ЭВМ КА – 16 тыс. строк.

Разработка программного обеспечения «УСК-МО» завершился в 1990-м году переходом на этап эксплуатации и сопровождения. В 1996-м году система первого этапа принята на вооружение.

6.4. Создание бортовых программных комплексов стратегических ракет и космических аппаратов

Примером одной из организаций в СССР, создававших бортовые системы ЭВМ и комплексы программ управления для ракет и космических аппаратов, являлось ***Харьковское научно-производственное объединение «Хартрон»,*** созданное в 1959-м году. Более 40 лет оно оставалось ведущим разработчиком систем управления, бортовых и наземных вычислительных комплексов, сложного электронного оборудования для различных типов ракет и космических аппаратов. Им созданы комплексы программ систем управления межконтинентальных баллистических ракет СС-7, СС-8, СС-9, СС-15, СС-18, СС-19, самой мощной в мире ракеты носителя Энергия, орбитальных модулей «Квант», «Кристалл», «Природа», «Спектр», 152 спутников серии «Космос» и других объектов [9, 11].

Первые системы управления строились с аналоговыми приборами систем стабилизации, а с 1964-го года электронными счетно-решающими приборами. На этапе создания и последующего выпуска электронных счетно-решающих приборов в Научно-производственном объединении «Хартрон» было организовано современное и мощное производство модулей, многослойных печатных плат, запоминающих устройств на ферритовых сердечниках, решены сложные научно-технические проблемы обеспечения помехозащищенности, высокой надежности, стабильности параметров бортовой вычислительной техники длительного срока эксплуатации.

Генеральным директором и Главным конструктором систем управления для ракетных комплексов в научно-производственном объединении «Хартрон» с 1960 – го по 1986-й год был Владимир Григорьевич Сергеев. К середине 60-х годов стало ясно, что принцип построения систем управления на основе аналоговых и дискретных счётно-решающих устройств не имеет перспективы. Дальнейшее совершенствование управления межконтинентальными баллистическими ракетами требовало резкого увеличения объёмов информации, обрабатываемой на борту ракеты в реальном масштабе времени. Требовалось также принципиально изменить идеологию регламентных проверок систем ракеты, которая базировалась на использовании сложной, наземной, дорогой и неудобной в эксплуатации передвижной испытательной аппаратуры, размещаемой в кузовах нескольких

автомобилей.

Революционным шагом на этом этапе явилось использование в системах управления ракет ***бортовых электронных вычислительных машин,*** обеспечивающих функционирование ракетного комплекса при наземных проверках и в условиях полета ракеты. При этом резко упрощалась наземная аппаратура, ее можно было разместить в ракетных шахтах, отказавшись от автопоездов. Возможность решения более сложных алгоритмов позволяла существенно повысить точность стрельбы.

В предприятии, было создано подразделение (Борис Михайлович Конорев) по определению требований к архитектуре и вычислительным характеристикам бортовых ЭВМ и разработке программного продукта. Потребовалось создать не только новую методологию разработки всех алгоритмов и программ полета, наземных испытаний, но и создавать новую технологию проектирования технических средств, включая моделирующие стенды, систему автоматизированного производства комплексов программ.

Вначале создание систем управления с бортовыми ЭВМ в «Хартроне» шло по двум направлениям:

• применение бортовой ЭВМ, разработанной головным предприятием по вычислительной технике Министерства радиопромышленности СССР – Научно исследовательским центром вычислительной техники;

• использование бортовой ЭВМ собственной разработки.

В апреле 1967-го года Генеральный директор и Главный конструктор Владимир Григорьевич Сергеев предложил обсудить и решить вопрос о концентрации сил на одном из этих направлений. Все руководители ведущих подразделений: высказались за использование бортовой ЭВМ собственной разработки, поскольку в ***«чужую» машину*** было практически невозможно оперативно вносить необходимые изменения в аппаратуру и программы, что резко замедлило бы разработку новых компонентов систем управления. Уже в 1968-м году был испытан первый экспериментальный образец бортовой ЭВМ на гибридных модулях. В 1971-м году, впервые в СССР был произведен запуск новой ракеты 15А14 с системой управления, включающей бортовую ЭВМ.

Удачно выбранный и успешно реализованный комплекс вычислительных характеристик ЭВМ (разрядность 16, объём ОЗУ 512-1024 слов, объём ПЗУ 16 слов, быстродействие 100 тыс. опер/сек.), надежная элементная база обеспечили этой бортовой ЭВМ уникальный срок жизни – около 25 лет, а ее несколько модернизированный вариант находится в эксплуатации на боевом

дежурстве. В целях обеспечения малых габаритно-массовых характеристик ЭВМ впервые в отрасли были созданы гибридные микросборки схем управления оперативным запоминающим устройством, плоские микромодули согласующих устройств с гальванической развязкой, многослойные печатные платы, изготовленные методом открытых контактных площадок.

В 1979-м году были приняты на вооружение ракеты 15А18 и 15А35 с унифицированным бортовым вычислительным комплексом. Для систем управления этих изделий, была разработана новая технология отработки программно-математического обеспечения, включающая так называемый ***«электронный пуск»,*** при котором на специальном ***имитационном моделирующем комплексе,*** включающем ЭВМ БЭСМ-6 и изготовленные блоки системы управления ракетой, моделировался полет ракеты и реакция системы управления на воздействие основных возмущающих факторов. Эта технология обеспечила также эффективный и полный контроль полетных заданий.

В последующие годы, были созданы еще четыре поколения бортовых ЭВМ, имеющих одни из лучших Советском Союзе вычислительные и эксплуатационные характеристики, а также разработана эффективная технология производства программных комплексов. Одной из «изюминок» была оригинальная система динамической коррекции программ. Именно она обеспечила возможность (при наличии ПЗУ с жёсткой «прошивкой» программ с помощью «косичек», вставляемых в П-образные ферритовые сердечники) оперативного внесения необходимых изменений в программный комплекс на всех этапах работ от предстартовых испытаний до работы на орбите.

Опыт эксплуатации первых бортовых ЭВМ показал настоятельную необходимость совершенствования структурных методов повышения надежности. Учеными и инженерами предприятия, были разработаны теоретические основы синтеза высоконадёжных, вычислительных структур, с многоярусным мажоритированием и адаптацией. Они легли в основу последующих поколений бортовых ЭВМ.

В 1984-м – 88-м годы была создана и отработана система управления для уникальной мощной ракеты СС18, известной по зарубежной классификации как ***«Сатана».*** В этой разработке были успешно внедрены все лучшие технические решения, наработанные на предшествующих заказах, а также целый ряд принципиально новых идей:

• обеспечение работоспособности после воздействия ядерного взрыва в полёте;

• высокоточное индивидуальное разведение боевых блоков;

• прямой метод наведения не требующий ранее подготовленного полетного задания;

• обеспечение дистанционного нацеливания.

Решение этих задач обеспечивалось новым мощным ***бортовым вычислительным комплексом*** с использованием полупроводниковых «пережигаемых» постоянных и электронных оперативных запоминающих устройств. Основная элементная база разрабатывалась и изготавливалась в Минском производственном объединении Интеграл и обеспечивала необходимый уровень радиационной стойкости. Для одного из видов боевых блоков было разработано и впервые в Советском Союзе прошло летные испытания запоминающее устройство на цилиндрических магнитных доменах.

Одной из самых сложных задач было создание бортового многомашинного вычислительного комплекса для ракеты-носителя «Энергия», решающего сложнейшие задачи стабилизации, выведения (с учетом нештатных ситуаций управления многочисленными двигательными установками), аварийной защиты двигателей, мягкой посадки спускаемых разгонных ступеней.

Высокие требования по надёжности и безотказности усугублялись использованием в ракете-носителе кислородных и водородных компонентов, что потребовало реализации в системе управления комплекса мер по обеспечению пожаро– и взрывобезопасности. Самой главной наградой за труд были два успешных запуска ракеты-носителя «Энергия» (22.02.1986 г. и 15.11.1988 г.) и успешное проведение натурных испытаний и сдача на вооружение ракеты СС-18.

Большой объём работ был проведен по созданию бортового вычислительного комплекса для ***систем управления космических аппаратов.*** Для летавших со станцией «Мир» модулей: Квант, Квант-2, Кристалл, Природа, Спектр, был создан комплекс с многоярусным мажоритированием, сохраняющий работоспособность при наличии 10–20 неисправностей. Опыт его безотказной эксплуатации на орбите в течение более 10 лет подтвердил правильность принятых технических решений. В конце 80-х годов для нового поколения систем управления космических аппаратов были созданы два бортовых вычислительных комплекса, имеющих, в отличие от предыдущих, существенно более низкое энергопотребление. Успешные запуски объектов, использующих эти комплексы, показали способность «Хартрона» обеспечивать космическую технику надежными бортовыми ЭВМ.

6.5. Космические аппараты информационных систем

На вычислительный центр (ВЦ) в Государственном союзном научно-исследовательском институте № 88 (ныне Центральный НИИ машиностроения Росавиакосмоса) и баллистическое подразделение НИИ-88 с 1963-го года возлагалась функция ***баллистического центра отрасли – БЦ-2*** (БЦ-1 – в Министерстве обороны, БЦ-3 – в Академии наук СССР). БЦ-2 начал участвовать ***в управлении космическими аппаратами (КА)*** [11]. Преобразование ВЦ в Координационно-вычислительный центр (КВЦ) с возложением на него информационного обеспечении государственных задач, по обработке и отображению информации при лётно-конструкторских испытаниях пилотируемых кораблей и автоматических станций, спутников научного и народнохозяйственного назначения. БЦ-2 стал головным по автоматическим межпланетным станциям («Венера», «Марс») и спутникам народнохозяйственного и научного назначения («Метеор», «Протон», АУОС), он выполнял функции дублирующего БЦ, по пилотируемым программам (корабли «Союз», станция «Салют»).

В 1959-м году было образовано ОАО ***«Информационные спутниковые системы»*** как восточный филиал ОКБ-1 Сергея Павловича Королёва в г. Красноярске-26 (ныне г. Железногорск Красноярского края) [11]. Его руководителем был назначен Михаил Федорович Решетнёв – ученик и соратник С.П. Королёва. Первый самостоятельный космический проект молодого коллектива – создание ракеты-носителя (РН) лёгкого класса типа «Космос». Предприятие заявило о себе 18 августа 1964-го года успешным пуском первой ракеты РН 11К65 («Космос-3») с тремя экспериментальными спутниками ***«Космос-38,-39,-40».***

ОАО ***«Информационные спутниковые системы» (ИСС) является примером*** успешной реализации ***массовых бортовых программных пролуктов космических систем*** во всех областях своей деятельности. С середины 60-х годов ОАО «ИСС» сохраняет ведущие позиции в России в области спутниковых телекоммуникаций и координатометрии. Оно является крупнейшим предприятием на Евразийском континенте по разработке, изготовлению, испытаниям и обеспечению эксплуатации космических комплексов и аппаратов на низких, средневысоких, высокоэллиптических и геостационарных орбитах. ОАО «ИСС» созданы и успешно эксплуатировались

большое количество космических аппаратов на орбитах высотой от 500 до 40000 км – это ***две трети отечественных спутников.*** В отдельные годы на различных орбитах одновременно работало свыше 120 спутников ОАО «ИСС», которые составляли фундамент национальной орбитальной группировки. На их основе создано свыше 30 космических систем и комплексов, обеспечивших ***национальные потребности в*** сферах:

- обороноспособности;
- информационной безопасности;
- развития народного хозяйства;
- решения социальных задач;
- развития культуры;
- международного сотрудничества;
- полномасштабного включения страны в мировое информационное сообщество.

Все космические системы содержали управляющие бортовые ЭВМ с программными продуктами размером в десятки и сотни тысяч строк текста каждый. Общий объем комплексов программ в реализованных космических аппаратах составляет многие миллионы строк. В начале 60 – х годов предприятие приступило к самостоятельной разработке космических аппаратов и спутниковых систем. С 1964-го года его коллективом впервые в стране были созданы и выведены на низкие орбиты малые спутники связи, с 1967-го года – навигационные и научные КА, с 1968-го года – геодезические спутники. С середины 60-х годов предприятие также начало осваивать производство самых мощных тогда космических аппаратов типа «Молния-1», запускаемых на высокую эллиптическую орбиту. В 1967-м году в космос были выведены три спутника «Молния-1», изготовленных в Сибири. На базе космических аппаратов этого типа впервые в мире была создана система связи и телевещания на высоких эллиптических орбитах.

С середины 70-х годов предприятие приступило к освоению геостационарной орбиты (ГСО) – самой коммерчески востребованной области околоземного пространства. С тех пор ОАО «ИСС» ***обеспечивает национальные интересы*** в области спутниковых телекоммуникаций на ГСО, поддерживает непрерывное функционирование и наращивание систем спутниковой связи, телевещания, ретрансляции. С конца 70-х и до середины 90-х годов ОАО «ИСС» создало геостационарные многоспутниковые группировки мощных КА различного назначения – «Горизонт», «Поток», «Луч», «Экран-М», «Радуга-1», «Экспресс», «Экспресс-АМ». В 1982-м году на орбиту был выведен первый космический аппарат глобальной навигационной

спутниковой системы ГЛОНАСС, разработанный в ОАО «ИСС». В 2003-м году в соответствии с Федеральной целевой программой предприятие приступило к ее модернизации современными КА «Глонасс-М».

Центр управления полетами (ЦУП) с новым комплексом технических средств создан для обеспечения реализации, совместного с США экспериментального проекта «Союз – Аполлон» [11]. Специалистами советского и американского ЦУПов проведена большая работа по согласованию терминологии, моделей движения КА, технической документации, организации взаимодействия двух ЦУПов. Обеспечено управление модернизированными кораблями «Союз» в беспилотном и пилотируемом вариантах. БЦ центра управления полетами – стал головным по пилотируемым программам. В ЦУПе создано подразделение командно-программного обеспечения управления полётом. Совместный полет пилотируемых кораблей «Союз» (СССР) и «Аполлон» (США) с их стыковкой на орбите и взаимными переходами членов экипажей из корабля в корабль состоялся в июле 1975-го года.

На ЦУП возложена задача по управлению полётами всех отечественных космических кораблей, пилотируемых орбитальных и автоматических межпланетных станций. Модернизированы технические средства для обеспечения длительной работы с пилотируемыми орбитальными станциями и управление полётом орбитальной станции «Салют-6». На станции побывало 27 человек, в том числе 8 иностранных космонавтов. К станции совершил полет и стыковался с ней 31 КА, в том числе 16 пилотируемых кораблей. Обеспечено управления полётом автоматических межпланетных станций. Полёты автоматических станций к Венере состоялись в 1978-м, 81-м и 1983-х годах, к Венере и комете Галлея – в 1984-м – 86-х годах, к Марсу и его спутнику Фобосу – в 1988-м– 89-х годах. Осуществлялось управление полётом орбитальной станции «Салют-7». На станции побывал 21 человек, в том числе 2 иностранных космонавта. К станции совершили полёты и стыковались с ней 24 КА, в том числе 9 пилотируемых кораблей.

Опираясь на высокий научный и кадровый потенциал, ОАО «Информационные спутниковые системы» создало мощную производственно-технологическую и экспериментальную базу. За эти годы специалисты предприятия изготовили множество космических аппаратов различного назначения, на основе которых был сформирован ряд многоспутниковых систем и комплексов на всех орбитах.

В рамках ***технологической платформы предприятия***

развиваются следующие группы технологий [11.

• создания информационно – вычислительных систем на основе концепции интегрированной модульной архитектуры для бортовых радиоэлектронных систем;

• создания автономных бесплатформенных инерциальных навигационных систем и их чувствительных элементов;

• создания бортовых систем управлении как высоко динамичными, так и слабо маневренными объектами на основе отказоустойчивых, многократно резервированных средств управления и контроля внешней среды;

• создания радиолокационных систем, в том числе с активными фазированными антенными решетками;

• формирования информационно – управляющего поля и бортовых экспертных систем на их основе;

• создания интеллектуальных датчиков первичных параметров, являющихся основой современных информационных систем;

• комплексная технология навигации, наблюдения и связи, обеспечивающая повышение безопасности и экономической эффективности транспортных систем;

Технологии, развиваемые в рамках технологической платформы (ТП), окажут воздействие, как непосредственно на традиционные приборостроительные проекты в сфере авиационного транспорта, так и на смежные отрасли экономики:

• водный (морской), железнодорожный, автомобильный транспорт;

• авиационная промышленность;

• ракетно-космическая промышленность;

• радиоэлектронная промышленность;

• транспортные перевозки (авиационные, железнодорожные, речные);

• оборонно-промышленный комплекс;

• образование.

Создание базового комплекса нового поколения предполагалось на основе системной технологии «Базовые интегрированные комплексы открытой архитектуры, на основе интегрированной модульной авионики». Использование концепции «базовый комплекс – семейство модификаций» позволит проводить поэтапное наращивание функций комплекса без изменения его архитектуры, за счет чего значительно экономить затраты и время на реализацию проектов или участвовать одновременно в нескольких полномасштабных проектах. В рамках указанного подхода основной продукцией будут интегрированные комплексы приборного оборудования.

6.6. Бортовые программные комплексы авиационных систем в 80-е годы

Первоначально основным потребителем авиационной электроники и программной инженерии были военные самолеты. ***Управление самолетом*** обеспечивали системы: связи; навигации; индикации; управления полетом; предупреждения столкновений; метеонаблюдения; управления самолетом; регистрации полета. В системы, обеспечивающие ***управление системами вооружения входили,*** радары; сонары; электронно-оптические системы; системы обнаружения целей; системы управления вооружением. Цифровые вычислительные машины в составе бортового оборудования самолетов появились на рубеже 60-х годов (см. п. 2.6.) и за относительно короткий срок практически полностью заменили использовавшиеся ранее аналоговые вычислители, поскольку обеспечивали более высокую точность решения задач, характеризовались большей универсальностью применения и обладали широкими логическими возможностями. Эти качества бортовой цифровой вычислительной машины (БЦВМ) позволяли ее использовать практически ***во всех подсистемах бортового оборудования самолета,*** обеспечивали устойчивость к усложнению алгоритмов и позволяли применять более сложные, а значит, и более совершенные законы управления самолетом и его подсистемами. Они позволили осуществить информационное взаимодействие между отдельными (ранее непосредственно не взаимодействовавшими) подсистемами бортового оборудования, что повысило эффективность выполнения полетного задания и безопасность полетов [11].

На ранних стадиях развития основное внимание специалистов уделялось разработке ЭВМ и средств ее сопряжения с бортовой аппаратурой. Проблема создания программных продуктов обострилась по мере усложнения структуры машин, расширения круга решаемых задач, появления и развития бортовых вычислительных систем, и в настоящее время затраты на разработку программных продуктов БЦВМ превышают затраты на создание аппаратных средств. Для производства комплексов программ начинали с использования языков уровня ассемблера, а для их отработки – специальные отладочные комплексы, объединяющие

БЦВМ с инструментальной вычислительной машиной. Взаимодействие с абонентами БЦВМ первого и второго поколений производилось через устройство сопряжения (УС),

которое содержало необходимый набор преобразователей «аналог – цифра» и «цифра – аналог», так как бортовая аппаратура имела в основном аналоговый интерфейс.

Примером применения программной инженерии в авиации в 80-е годы может служить освоение в отечественной авиационной промышленности зарубежного стандарта [11]– ***Соглашение по сертификации бортовых систем и оборудования в части программного обеспечения —*** Стандарт DO-178 «Software Consideration in Airborne Systems and Equipment Certifi-cation» разработан и поддерживается ассоциацией RTCA (Radio Technical Commission for Aeronautics). Первая его версия была принята за рубежом в 1982-м году, а вторая (DO-178A) – в 1985-м году, которая вскоре начала применяться на некоторых отечественных авиационных предприятиях. В 1992-м году была утверждена версия DO-178B, которая надолго стала базовой и представлена ниже. Стандартом предусмотрено пять уровней серьезности отказа программного продукта, и для каждого из них должен использоваться набор требований, которые должны гарантировать работоспособность и качество всей системы в целом, при возникновении отказов определенного уровня серьезности [11].

Пример процессов программной инженерии по авиационному стандарту DO-178B [11]:

«Соглашение по сертификации бортовых систем и оборудования в части программного обеспечения»

исчения»

	Стр.
1. ВВЕДЕНИЕ	1
1.1. Цель	1
1.2. Область действия	1
1.3. Связь с другими документами	1
1.4. Как использовать этот документ	1
1.5. Краткий обзор документа	3
2. СИСТЕМНЫЕ АСПЕКТЫ, СВЯЗАННЫЕ С РАЗРАБОТКОЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.	3
2.1. Поток информации между системой и процессами жизненного цикла ПО	4
2.1.1. Информационный поток от системных процессов к процессам программного обеспечения	4
2.1.2. Информационный поток от процессов программного обеспечения к системным процессам .	5
2.2. Отказные ситуации и уровень программного обеспечения	5
2.2.1. Классификация отказной ситуации .	6
2.2.2. Определения уровня программного обеспечения .	6
2.3. Проблемы архитектуры системы	8
2.3.1. Структуризация	8
2.3.2. Многоверсионное неидентичное программное обеспечение	9
2.3.3. Мониторинг безопасности	9
2.4. Системные вопросы модифицируемого пользователем программного обеспечения.	9

Программное обеспечение с возможностью выбора вариантов и готовое коммерческое программное обеспечение
2.5. Задачи проектирования систем для программного обеспечения, загружаемого в полевых условиях 10
2.6. Задачи системных требований по верификации программного обеспечения 11
2.7. Задачи программного обеспечения по верификации системы 12
3. ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ …… 12
3.1. Процессы жизненного цикла программного обеспечения 12
3.2. Определение жизненного цикла программного обеспечения 12
3.3. Критерии переходов между процессами 14
4. ПРОЦЕСС ПЛАНИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ . 15
4.1. Цели процесса планирования программного обеспечения 15
4.2. Работа процесса планирования программного обеспечения 15
4.3. Планы программного обеспечения 16
4.4. Планирование среды жизненного цикла программного обеспечения 17
4.4.1. Среда программирования 18
4.4.2. Язык программирования и компилятор 18
4.4.3. Среда тестирования программного обеспечения 19
4.5. Стандарты разработки программного обеспечения 19
4.6. Просмотр и обеспечение качества результатов процесса планирования программного обеспечения . 20

5. ПРОЦЕССЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 20
5.1. Процесс определения требований к программному обеспечению 21
5.1.1. Задачи процесса определения требований к программному обеспечению 21
5.1.2. Работы процесса определения требований к программному обеспечению 21
5.2. Процесс проектирования программного обеспечения 22
5.2.1. Задачи процесса проектирования программного обеспечения 22
5.2.2. Работы проектирования программного обеспечения 22
5.2.3. Проектирование модифицируемого пользователем программного обеспечения . 23
5.3. Процесс кодирования программного обеспечения 23
5.3.1. Задачи процесса кодирования программного обеспечения 23
5.3.2. Работы процесса кодирования программного обеспечения 23
5.4. Процесс интеграции 24
5.4.1. Задачи процесса интеграции 24
5.4.2. Работа процесса интеграции 24
5.4.3. Дополнительные вопросы интеграции 25
5.5. Прослеживаемость 25
6. ПРОЦЕСС ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ **25**
6.1. Цели процесса верификации программного обеспечения 26
6.2. Работа процесса ыерификации программного обеспечения 26
6.3. Просмотры и анализы программного обеспечения 27

6.3.1. Просмотры и анализы требований высокого уровня 27
6.3.2. Просмотры и анализы требований нижнего уровня 28
6.3.3. Просмотры и анализы архитектуры ПО 28
6.3.4. Просмотры и анализы исходного кода 29
6.3.5. Просмотры и анализы выходных данных процесса интеграции 30
6.3.6. Просмотры и анализы результатов и процедур тестовых наборов . 30
6.4. Процесс тестирования программного обеспечения 30
6.4.1. Среда тестирования 32
6.4.2. Выбор тестовых наборов, основанных на требованиях 33
6.4.2.1. Тестовые наборы для нормального диапазона 33
6.4.2.2. Тестовые наборы устойчивости к ошибкам 33
6.4.3. Методы тестирования, основанные на требованиях 34
6.4.4. Анализ тестового покрытия 35
6.4.4.1. Анализ тестового покрытия, основанного на требованиях 35
6.4.4.2. Анализ структурного покрытия 36
6.4.4.3. Выводы анализа структурного покрытия 36
7. ПРОЦЕСС УПРАВЛЕНИЯ КОНФИГУРАЦИЕЙ ПО **37**
7.1. Целевые функции процесса управления конфигурацией ПО 37
7.2. Работы процесса управления конфигурацией ПО 37
7.2.1. Идентиыикация конфигурации 37
7.2.2. Базы и трассируемость 38
7.2.3. Отчетность по дефектам. Трассировка и корректирующее действие 38
7.2.4. Управление изменениями 39

7.2.5. Проверка изменений 39
7.2.6. Учет состояния конфигурации 40
7.2.7. Архивирование, выборка и релиз . 40
7.2.8. Управление загрузкой ПО . 41
7.2.9. Управление окружением жизненного цикла ПО . 41
7.3. Категории управления данными 42
8. ПРОЦЕСС ОБЕСПЕЧЕНИЯ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 42
8.1. Задачи процесса обеспечения качества программного обеспечения 43
8.2. Работа процесса обеспечения качества программного обеспечения 44
8.3. Оценка согласованности программного обеспечения 44
9. ПРОЦЕСС СЕРТИФИКАЦИОННОГО ОТСЛЕЖИВАНИЯ 45
9.1. Средства согласования и планирования 45
9.2. Обоснование согласованности 45
9.3. Минимум данных жизненного цикла ПО, которые передаются сертификационному ведомству 45
9.4. Данные жизненного цикла ПО, относящиеся к типовому проекту 45
10. ДОКУМЕНТАЦИЯ, СОЗДАВАЕМАЯ В ПРОЦЕССЕ ЖИЗНЕННОГО ЦИКЛА ПО 46
10.1. План сертификации в части программного обеспечения 47
10.2. План разработки ПО 48
10.3. План верификации ПО 48
10.4. План управления конфигурацией ПО 49
10.5. План обеспечения качества программного обеспечения 50
10.6. Стандарты и требования к программному обеспечению 51
10.7. Стандарты проектирования ПО 51

10.8. Стандарты кодирования ПО 52
10.9. Содержание требований к программному обеспечению 52
10.10. Описания проекта 52
10.11. Исходный текст 53
10.12. Исполняемый объектный код . 53
10.13. Процедуры и директивы верификации программного обеспечения 53
10.14. Результаты верификации программного обеспечения 54
10.15. Указатель конфигурации среды жизненного цикла ПО 54
10.16. Указатель конфигурации ПО 54
10.17. Отчеты об ошибках 55
10.18. Протоколы управления конфигурацией ПО 55
10.19. Протоколы обеспечения качества ПО 55
10.20. Итоговый документ разработки программного обеспечения 55
11. ДОПОЛНИТЕЛЬНЫЕ ВОПРОСЫ 57
11.1. Использование ранее разработанного программного обеспечения 57
11.1.1. Модификация ранее разработанного программного обеспечения . 57
11.1.2. Изменение системы или объекта 57
11.1.3. Изменения среды применения или среды разработки 58
11.1.4. Изменение базовой версии разработки 59
11.1.5. Вопросы управления конфигурацией программного обеспечения 59
11.1.6. Вопросы обеспечения качества программного обеспечения 59
11.2. Аттестация инструментальных средств 60
11.2.1. Критерий аттестации для средств разработки ПО 61
11.2.2. Критерий аттестации для средств верификации ПО 62

11.2.3. Отчеты по аттестации средств 62
11.2.3.1. План аттестации средств 62
11.2.3.2. Эксплуатационные требования для средств 62
11.2.4. Соглашение об аттестации средств 63
11.3. Альтернативные методы 63
11.3.1. Формальные методы 63
11.3.2. Исчерпывающее тестирование входных данных 64
11.3.3. Вопросы верификации для многоверсионного неидентичного программного обеспечения 65
11.3.3.1. Независимость многоверсионного неидентичного программного обеспечения 66
11.3.3.2. Верификация для случая нескольких процессов 66
11.3.3.3. Верификация многоверсионного исходного кода 66
11.3.3.4. Аттестация инструментальных средств для многоверсионного неидентичного программного обеспечения 66

Представленные требования стандарта DO-178B охватывают основные ***технологические процессы программной инженерии*** проектирования и производства программных продуктов для гражданских и военных самолетов, а также для других сложных систем реального времени. Это первый систематизированный международный документ, который в конце 1980-х годов ***заложил фундамент*** для широкого развития стандартизации методов и документов программной инженерии, организациями ISO, IEC, IEEE. Стандарт был переведен на русский язык, и активно применялся отечественной промышленностью и способствовал повышению культуры производства программных продуктов.

Заключение

В 1960-е– 80-е годы были решены и практически апробированы основные ***научно-технические проблемы создания на ЭВМ сложных программных продуктов реального времени высокого качества.*** Разработаны методы и регламенты организации труда больших коллективов специалистов разной квалификации, для проектирования, производства, сопровождения и обеспечения длительного жизненного цикла программных продуктов сложных систем. Одновременно и независимо многие подобные проблемы решались на ряде отечественных предприятий для высокоточного технологического производства, динамического управления объектами и процессами, вычислительными средствами и комплексами программ для авиационных, морских, космических и других систем оборонного назначения. ***Индустриальные принципы проектирования и производства сложных технических систем*** стали использоваться для создания современных заказных программных продуктов управления и обработки информации систем реального времени. Основные промышленные методы производства сложных вычислительных систем оборонного назначения и крупномасштабного программирования стали ***базовыми для программной инженерии*** и ***интеллектуальной основой современных технических систем в различных областях.***

Программные продукты оказались одними из наиболее сложных ***интеллектуальных*** компонентов технических систем высокого качества, создаваемых человеком. Сформировался ***новый вид промышленности– специфическая наука, методология и технология производства – программная инженерия.*** Полное отсутствие информации о подобных разработках на Западе и на соседних оборонных предприятиях обеспечило оригинальность решений, которые, в ряде случаев, впоследствии оказались универсальными и принципиальными для вычислительных систем реального времени различного назначения. В ***эти годы*** в результате эффективной разработки программных продуктов для ***отечественных оборонных систем они имели функциональные характеристики и качество решения задач, практически адекватные аналогичным зарубежным оборонным системам, и тем самым обеспечивали паритет в обороноспособности нашей страны.***

Для эффективного проектирования и производства программных продуктов высокого качества стала

необходимой профессиональная подготовка квалифицированных коллективов руководителей и специалистов, освоение ими современных методов анализа, оценивания и прогнозирования процессов и необходимых ресурсов при создании комплексов программ. Основной интеллектуальный труд специалистов вкладывается в разработку функциональных алгоритмов и текстов программ, а также в их интеграцию, испытания, документирование на всех этапах проектирования и производства сложных программных продуктов. Следует выделять и обучать руководителей и специалистов, ***ответственных*** за соблюдение современных ***экономически эффективных технологических методов и стандартов*** промышленного создания и развития сложных, заказных программных комплексов ***гарантированного высокого качества.***

Ключевой проблемой на всех стадиях проектирования и производства заказных программных продуктов является ***предотвращение, диагностика, выявление и устранение дефектов и ошибок.*** Каждая ошибка в программном продукте может стимулировать отказы функционирования и ***даже катастрофы систем.*** Простейшие ошибки в программных продуктах зачастую приводят к авариям и катастрофам сложных систем с гибелью людей и дорогой техники. В нашей стране и в мире зарегистрировано множество случаев трагических катастроф в авиации и ракетной технике, причинами которых были элементарные ошибки в программах мобильных и бортовых ЭВМ. Прогнозирование и достижение высокого качества комплексов программ однозначно связано с предотвращением, регистрацией и ликвидацией причин возможных катастроф при их функционировании. Следует учитывать, что затраты на обнаружение и устранение дефектов быстро возрастают при приближении к завершающим этапам испытаний и сертификации сложных программных продуктов, прогнозирование и ликвидация ошибок должна быть ***доминирующей задачей обучения*** всех руководителей и специалистов каждого заказного проекта.

Гарантирование качества продукции осуществляется посредством сертификационных испытаний ***процессов производства*** комплексов программ, а также испытаний их результатов – ***программных продуктов.*** Процедуры ***сертификации*** должны подтверждать соответствие требованиям заказчика, посредством которых независимое от изготовителя и потребителя предприятие ***юридически удостоверяет*** в письменной форме, что состояние продукции и качество функционирования способно обеспечить стабильность характеристик изготовляемой продукции и соответствует

установленным заказчиком требованиям к функциям, характеристикам качества и стандартам.

Для этого они должны проходить тщательные испытания в условиях их последующего применения, ***динамически имитирующих реальную внешнюю среду*** для генерации тестов и функционирования испытываемых комплексов программ. ***Требования и результаты реализация тестов должны полностью отражать возможность проверки утвержденных заказчиком требований к программному продукту,*** и соответственно, по сложности и трудоемкости, должны быть принципиально аналогичными разработке такого программного комплекса. Эти ***два представления программного продукта*** отличаются только формой описания их содержания: функциональным (процессным) или событийным (сценариями и результатами исполнения). Таким образом, формируется возможность выполнять испытания, выявлять и устранять ошибки, предотвращать катастрофы систем, а также ***достигать высокого уровня соответствия программного комплекса заданным требованиям.***

Кроме сертификации технологических процессов и готовых программных продуктов для эффективного их производства, и применения в промышленности, необходима ***сертификация квалификации коллектива специалистов,*** реализующих эти процессы. Для этого следует их обучение и аттестация на допуск к участию в таких работах, требующих высоких уровней профессиональной и интеллектуальной квалификации. Они должны освоить, знать и уметь использовать методологию программной инженерии, методы программирования, верификации, тестирования и документирования программных компонентов и комплексов, а также основы сертификации производственных процессов и программных продуктов. Квалифицированным специалистам в промышленности следует учитывать ***юридическую ответственность*** за качество производства и продуктов, за результаты сертификации и достоверность документации при применении программных продуктов пользователями.

Сертификаты специалистов программных продуктов должны демонстрировать и создавать уверенность у руководителей и заказчиков, что они способны квалифицированно и своевременно выполнять порученные работы.

Заказные программные продукты должны проходить ***экономическое обоснование*** и прогнозирование выбора и применения комплексной системы качества, методологии и

инструментальных средств, ***гарантирующих требуемые качество,*** надежность и безопасность функционирования программного продукта. ***Предсказуемые высокие экономические характеристик, качество производства компонентов,*** и программного продукта, должны сопровождать весь жизненный цикл сложных комплексов программ. ***Развитие экономики*** в этой области индустрии ***связано с большими профессиональными и психологическими трудностями,*** типичными для новых разделов современной науки и техники, проявляющимися на стыке различных областей методов и знаний. Широкий спектр технических характеристик таких объектов, количественных и качественных факторов, которые, с различных сторон, характеризуют содержание компонентов и комплексов программ, а также невысокая достоверность экономической оценки их значений, определяют значительную ***трудность при описании и измерении*** свойств, количества и качества сложных программных продуктов для их экономической оценки и прогнозирования характеристик качества.

Разработка больших комплексов программ реального времени ***в гражданских областях*** промышленности и применения ЭВМ в 1960-е – 70-е годы ***не была востребована.*** Это определялось ориентацией научных и учебных учреждений, а также предприятий гражданской промышленности на создание относительно небольших расчетных программ и ***способствовало их отставанию в индустрии проектирования крупных комплексов программ реального времени почти на десяток лет от оборонной промышленности.*** Только в 1980-е годы, при быстром прогрессе в микропроцессорной технике появилась возможность, и было востребовано создание крупных заказных программных продуктов для административных, финансовых, промышленных и иных систем. Однако положительный опыт и методология оборонной промышленности по ***программной инженерии*** и индустриальному созданию крупных комплексов программ оставались секретными, а методы разработки небольших программ оказались не применимыми или не эффективными. В дальнейшем многие отечественные специалисты и предприятия были вынуждены заново ***осваивать и развивать методологию индустриального проектирования и производства крупных заказных программных продуктов***, в том числе с оглядкой на опыт оборонной промышленности прошлых лет.

В последние годы было разработано свыше ***50-ти международных стандартов*** ISO, IEC, обеспечивающих систематизированный выбор необходимых требований и

процессов, в зависимости от характеристик, функций и особенностей конкретных проектов, а также формирование на их базе проблемно-ориентированных профилей стандартов для определенных типов проектов и предприятий. Практическое применение стандартов, ***сосредоточивших мировой опыт*** создания различных крупных комплексов программ, способствовал значительному повышению производительности труда специалистов и качества создаваемых программных продуктов. В результате, оформились базовые понятия, процессы и нормативные документы, ***современной научно-технической и промышленной области знаний и применения программной инженерии.*** Таким образом, к настоящему времени существуют методологические и нормативные ***документы*** для освоения, применения и развития ***современной инженерной дисциплины*** создания крупных программных продуктов высокого качества.

Для отечественных специалистов ***ориентиром*** может служить, приведенный на задней обложке цикл современных учебников и монографий по программной инженерии. Опубликованы многочисленные (в основном переводные) монографии и учебники для подготовки специалистов разного уровня. В качестве основы для разработки учебных планов могут рассматриваться: международный стандарт – Свод знаний по программной инженерии – ISO 19759:2005 – ТО. – SWEBOK, а также ***«Рекомендации по преподаванию программной инженерии и информатики*** в университетах – SE2004 (пер. с англ. – М. 2007)». В 2011-м году утвержден Государственный ***образовательный стандарт 23000 Программная инженерия*** (см. приложение).

Приложение

Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки 231000 программная инженерия (квалификация (степень) «магистр»)

(Основные положения)

Стандарт высшего профессионального образования (ФГОС ВПО) представляет собой совокупность требований, обязательных при реализации основных образовательных программ магистратуры по направлению подготовки 231000 Программная инженерия всеми образовательными учреждениями высшего профессионального образования (высшими учебными заведениями) на территории Российской Федерации. Право на реализацию основных образовательных программ высшее учебное заведение имеет только при наличии соответствующей лицензии, выданной уполномоченным федеральным органом исполнительной власти.

Характеристика профессиональной деятельности магистров

Областью профессиональной деятельности магистров по направлению подготовки 231000 Программная инженерия является индустриальное производство программного обеспечения для информационно-вычислительных систем различного назначения. Объектами профессиональной деятельности выпускников по направлению 231000 Программная инженерия являются:

- программный проект (проект разработки программного продукта);
- программный продукт (создаваемое программное обеспечение);
- процессы жизненного цикла программного продукта;
- методы и инструменты разработки программного продукта;
- персонал, участвующий в процессах жизненного цикла.

Видами профессиональной деятельности выпускников являются:

- научно-исследовательская;
- аналитическая;
- проектная;
- технологическая;
- производственная;

• педагогическая;

• организационно-управленческая;

• сервисно-эксплуатационная.

Конкретные виды профессиональной деятельности, к которым в основном готовится магистр, определяются высшим учебным заведением совместно с обучающимися, научно-педагогическими работниками высшего учебного заведения и объединениями работодателей.

Магистр по направлению подготовки 231000 Программная инженерия должен быть подготовлен к решению следующих профессиональных задач в соответствии с профильной направленностью магистерской программы и видами профессиональной деятельности:

научно-исследовательская деятельность

• разработка методов исследования объектов профессиональной деятельности на основе общих тенденций развития программной инженерии;

• оптимизация проектных и технологических решений с целью обеспечения качества объектов профессиональной деятельности;

• организация научно-исследовательской работы;

аналитическая деятельность:

планирование, управление и контроль выполнения требований;

• оценки степени трудности, рисков, бюджета и времени в течение выполнения проекта, контроль рабочего графика;

проектная деятельность:

проектная деятельность в профессиональной сфере на основе системного подхода, построение и использование моделей, осуществление их качественного и количественного анализа;

• формирование технических заданий и руководство разработкой программного обеспечения;

• выбор методологии проектирования объектов профессиональной деятельности;

технологическая деятельность:

применение современных технологий разработки программных комплексов с использованием автоматизированных систем планирования и управления, контроль качества разрабатываемых программных продуктов;

производственная деятельность:

планирование и руководство процессом разработки программного обеспечения;

обучение и аттестация пользователей программного обеспечения;

организационно-управленческаядеятельность:

разработка технических заданий и проведение технико-экономического обоснования;

• организация работы коллектива разработчиков программного продукта, осуществление взаимодействия со смежниками;

сервисно-эксплуатационная деятельность:

выбор технической и экономической моделей эволюции и сопровождения программного обеспечения.

Выпускник должен обладать следующими ***общекультурными компетенциями:***

способность совершенствовать и развивать свой интеллектуальный и общекультурный уровень;

• способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности;

• умение свободно пользоваться русским и иностранным языками как средством делового общения;

• использование на практике умения и навыки в организации исследовательских и проектных работ, в управлении коллективом;

• способность проявлять инициативу, в том числе в ситуациях риска брать на себя всю полноту ответственности;

• способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности;

• способность к профессиональной эксплуатации современного оборудования и приборов (в соответствии с целями магистерской программы).

Выпускник должен обладать следующими профессиональными компетенциями ***научно-исследовательской деятельности:***

умение отбирать и разрабатывать методы исследования объектов профессиональной деятельности на основе общих тенденций развития программной инженерии;

• умение проводить анализ, синтез, оптимизацию решений с целью обеспечения качества объектов профессиональной деятельности;

• умение организовывать самостоятельную и коллективную научно-исследовательскую работу.

Аналитическая деятельности:

умение планировать, управлять и контролировать выполнение требований;

• умение выполнять оценки степени трудности, рисков, бюджета и времени в течение выполнения проекта, осуществлять контроль рабочего графика.

Проектная деятельность:

способность к проектной деятельности в профессиональной сфере на основе системного подхода, умение строить и использовать модели для описания и прогнозирования различных явлений, осуществлять их качественный и количественный анализ;

• умение формировать технические задания и способность руководить разработкой программного обеспечения;

• умение оценить и выбрать методологию проектирования объектов профессиональной деятельности.

• умение применять современные технологии разработки программных комплексов с использованием автоматизированных систем планирования и управления, осуществлять контроль качества разрабатываемых программных продуктов.

Производственная деятельность:

умение планировать и осуществлять руководство процессом разработки программного обеспечения.

Педагогическая деятельность:

готовность использовать современные психолого-педагогические методы в профессиональной деятельности;

• способность использовать педагогические приемы, принципы обучения и аттестации пользователей программного продукта при организации обучения;

• навыки подготовки и проведения учебных занятий по дисциплинам направления «Программная инженерия».

Организационно-управленческая деятельность:

способность рассчитывать и оценивать условия и последствия принимаемых организационно-управленческих решений;

• умение разработать техническое задание и провести технико-экономическое обоснование;

• способность организовывать работу коллектива разработчиков программного продукта, умение осуществлять взаимодействие со смежниками.

Сервисно-эксплуатационная деятельность:

умение осуществлять выбор технической и экономической моделей эволюции и сопровождения программного обеспечения.

Реализация компетентностного подхода должна предусматривать широкое использование в учебном процессе активных и интерактивных форм проведения занятий (семинаров

в диалоговом режиме, дискуссий, компьютерных симуляций, деловых и ролевых игр, разбора конкретных ситуаций, психологических и иных тренингов, групповых дискуссий, результатов работы студенческих исследовательских групп, вузовских и межвузовских телеконференций) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Высшее учебное заведение обязано обеспечивать гарантию качества подготовки, в том числе путем:

• разработки стратегии по обеспечению качества подготовки выпускников с привлечением представителей работодателей;

• мониторинга, периодического рецензирования образовательных программ;

• разработки объективных процедур оценки уровня знаний и умений обучающихся, компетенций выпускников;

• обеспечения компетентности преподавательского состава;

• регулярном проведении самообследования по согласованным критериям для оценки своей деятельности (стратегии) и сопоставления с другими образовательными учреждениями с привлечением представителей работодателей;

• информирования общественности о результатах своей деятельности, планах, инновациях.

Оценка качества освоения магистерских программ должна включать текущий контроль успеваемости, промежуточную аттестацию обучающихся и итоговую государственную аттестацию выпускников.

Литература

1. Ершов А.П., Шура-Бура М.Р. Становление программирования в СССР. Препринты ВЦ СОАН СССР № 12, 13. 1976.

2. Лебедев С.А. К 100-летию со дня рождения основоположника отечественной электронной вычислительной техники. – М.: ФИЗМАТЛИТ. 2002.

3. Малиновский Б.Н. История вычислительной техники в лицах. Киев. 1995.

4. Очерки истории информатики в России. Сб. сост. Д.А. Поспелов, А.Я. Фет. Новосибирск. Изд. СО РАН. 1998.

5. Китов А.И., Криницкий Н. А. Электронные цифровые машины и программирование. – М.: Гизфизматлит. 1959.

6. Ершов А.П. О некоторых человеческих и эстетических факторах в программировании. Кибернетика. № 5. 1972

7. Королев Л.Н., Томилин А.Н. С.А. Лебедев и развитие математического и программного обеспечения вычислительных машин СССР. Сб. Лебедев С.А. К 100-летию со дня рождения основоположника отечественной электронной вычислительной техники. – М.: ФИЗМАТЛИТ. 2002.

8. Бурцев В.С. Научная школа академика С.А. Лебедева в развитии вычислительной техники. Сб. Лебедев С.А. К 100-летию со дня рождения основоположника отечественной электронной вычислительной техники. – М.: ФИЗМАТЛИТ. 2002.

9. Первов М.А. Системы ракетно-космической обороны России создавались так – М.: АвиаРус-21. 2004.

10. Виртуальный компьютерный музей. – www.computer-museum.ru.

11. Интернет – Истории оборонных программных систем.

12. Радиопромышленность. Специальный выпуск. 70 лет МНИИПА. – М.: НИИЭИР. 2002.

13. «КОМЕТА» – 35 лет. – М.: «Оружие и технологии», 2008.

14. Власко-Власов К.А. От «Кометы» до «ОКО». Изд. КОМЕТА. 2005.

15. Боэм Б.У. Инженерное проектирование программного обеспечения. Пер. с англ. /Под ред. А.А. Красилова. – М.: Радио и связь, 1985.

16. Липаев В.В. Проектирование математического обеспечения АСУ. – М.: Советское радио, 32,5 а.л., 18 тыс. экз. 1977.

17. Липаев В.В. Качество программного обеспечения. —
М.: Финансы и статистика, 1983.

18. Липаев В.В., Серебровский Л.А., Гаганов П.Г., Штрик А.А. и др. Технология проектирования комплексов программ АСУ. – М.: Радио и связь, 1983.

19. Липаев В.В. Тестирование программ. – М.: Радио и связь, 1986.

20. Липаев В.В., Потапов А.И. Оценка затрат на разработку программных средств. – М.: Финансы и статистика, 1988.

21. Липаев В.В. Отечественная программная инженерия: фрагменты истории и проблемы. – М.: СИНТЕГ. 2007.

22. Липаев В.В. Программная инженерия. Методологические основы. – М.: Изд. Высшей школы экономики. 2006.

23. Пржиялковский В.В. Исторический обзор семейства ЕС ЭВМ. Виртуальный компьютерный музей. – www.computer-museum.ru.

24. Пржиялковский В.В. Операционные системы ЕС ЭВМ. Виртуальный компьютерный музей. – www.computer-museum.ru.

25. Левин В.К. Очерк становления Единой системы ЭВМ. Виртуальный компьютерный музей. – www.computer-museum.ru.

26. Чесноков В.В., Штейнберг В.И. Бортовые ЭВМ комплекса «Аргон». Виртуальный компьютерный музей. – www.computer-museum.ru.

27. Филинов Е.Н. Система малых ЭВМ (СМ ЭВМ). Виртуальный компьютерный музей. – www.computer-museum.ru.

28. История отечественных управляющих вычислительных машин (1955–1987 г.г.) Под редакцией д.т.н., профессора Я.А. Хетагурова.

Доктор технических наук. профессор, главный научный сотрудник Института системного программирования РАН, Заслуженный деятель науки и техники РСФСР, Лауреат премии Совета министров СССР, Лауреат премии Правительства РФ в области образования.

Владимир Васильевич ЛИПАЕВ

С 1954 по 1988 год работал в Московском НИИ приборной автоматики. До 1988 года – главный конструктор Министерства радиопромышленности С С СР по автоматизации проектирования программных средств, по технологии создания крупномасштабных программных продуктов для оборонных систем реального времени.

Многие годы занимается исследованиями и разработкой программных комплексов для систем обработки радиолокационной информации, методов и инструментальных средств для создания сложных управляющих программных продуктов реального времени высокого качества. Под его руководством разработан ряд больших инструментальных системы программной инженерии для автоматизации технологических процессов жизненного цикла сложных комплексов программ, широко использовавшихся в оборонной промышленности.

Автор более 50 монографий в области методов, технологий, инструментальных средств, тестирования и испытаний, стандартизации и сертификации проектирования и производства сложных комплексов программ. В последние годы опубликовал цикл учебников и монографий для вузов по программной инженерии, по методам и процессам промышленного производства крупных программных продуктов реального времени.